OS/2 Warp V4
in Team



Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Team OS/2:

OS/2 Warp V4 in Team

Vaterstetten, 1996

ISBN 3-932311-02-7

Primelstraße 23, 85591 Vaterstetten

1. Auflage

Coverdesign: Hawa & Nöh, Kassel

Druck: Druckerei Kösel, Kempten

ISBN 3-932311-02-7

Inhalt

Vorwort des Herausgebers

OS/2 lebt – und das trotz aller Unkenrufe, die seit Jahren seinen Niedergang prophezeit haben. Wirft man einen Blick auf die Entwicklungsgeschichte von OS/2, wird aber jedem schnell klar, daß OS/2 mit seiner Ausrichtung und seinen Fähigkeiten anderen PC-Betriebssystemen immer mindestens einen Schritt voraus war und das heute auch noch ist.

OS/2 war das erste 32-Bit-Betriebssystem für PCs, das Multitasking erlaubte, lange Dateinamen zuließ (HPFS) und im Kern sehr stabil war. Das schlägt sich noch heute in den Verkaufszahlen im Banken- und Versicherungssektor deutlich nieder, da sich diese Branchen auf ein zuverlässiges System verlassen können müssen.

Mit der Version 2.0 fügte IBM Anfang 1992 dem System eine leistungsfähige, objektorientierte Oberfläche zu, die im PC-Sektor immer noch ihresgleichen sucht. Damals wurde OS/2 als die »Integrationsplattform« bezeichnet, da neben der Möglichkeit, OS/2- und DOS-Programme laufen zu lassen, auch Windows-Anwendungen unter OS/2 liefen (und nicht selten sogar besser als unter dem echtem Windows).

Mitte 1993 wurde dann OS/2 2.1 ausgeliefert. Neben Verbesserungen an der 32-Bit-Graphics-Engine kamen auch die Unterstützung von Windows 3.1 und die Einführung des Multimedia Presentation Managers (MMPM) für OS/2 als fester Bestandteil des Betriebssystems hinzu.

Danach ging es mit Warp-Geschwindigkeit weiter: OS/2 Warp Version 3 wurde Ende 1994 ausgeliefert. Neben einigen Neuerungen, wie beispielsweise der Klickstartleiste, wurde der Hauptspeicherbedarf reduziert und das System von verbliebenem 16-Bit-Code befreit. Das Ergebnis war ein »echtes« 32-Bit-Betriebssystem, das die Prozessoren richtig ausnutzen konnte. Als Zugabe gab es eine BonusPak-CD, die neben einer kleinen Office-Suite auch Fax-, Multimedia-Programme sowie Internetzugangssoftware enthielt; alles in Form von echter 32-Bit-Software.

Während auf der CeBit 1995 das Publikum noch OS/2 Warp 3 bestaunte und das Betriebssystem mit Preisen für Design und Technik überschüttet wurde, lag schon wenig später in IBMs Schublade der Nachfolger Warp Connect bereit.

OS/2 Warp Connect Version 3 (so der vollständige Name) stellte sich als universeller Netzwerk-Client vor: Neben den Zugängen zu IBM LAN-Server, WfW, Windows 95 und Windows NT ermöglichte es den Zugriff auf Novell-Netze sowie den Zugang über TCP/IP zur Unix-Welt (Internet inklusive). Damit war OS/2 Warp Connect die ideale Basis für den vernetzten Arbeitsplatz.

Mit der gerade erschienenen Version 4 von OS/2 Warp demonstriert IBM gelungen seinen Anspruch auf die technische Vorherrschaft. Neben Verbesserungen am Kernel, der Netzwerk-Unterstützung und dem Hinzufügen neuer Technologien wie OpenGL und OpenDoc, sind es vor allem Techniken wie VoiceType, die einen immer wieder in Erstaunen versetzen.

Die vielen verschiedenen Funktionen von OS/2 sind inzwischen so zahlreich und komplex geworden, daß ein Autor alleine dieses Thema kaum allumfassend und dennoch kompetent abhandeln kann. Was lag also näher als die Idee, absolute Spezialisten ein Buch über OS/2 Warp Version 4 schreiben zu lassen?

Das Team OS/2 Deutschland wurde 1992 gegründet und ist eine von der IBM unabhängige Vereinigung von engagierten OS/2-Anwendern. Zum Zeitpunkt der Drucklegung dieses Buchs hat es mehr als 700 Mitglieder, die sich beruflich oder in ihrer Freizeit mit OS/2 auseinandersetzen, wobei jeder sein Spezialgebiet hat. Das Ergebnis dieser Idee halten Sie gerade in den Händen, ein Buch, zu dem jeder Autor sein spezielles Fachwissen gegeben hat, und das daher alle Ihre Fragen zu OS/2 beantworten wird.

Besonderer Dank gilt auch Thorsten Kowalski, der die CD zusammengestellt hat.

Ich wünsche Ihnen viel Erfolg mit dem neuen OS/2.

Bernd Rohrbach, 1996

Die Autoren

Oliver Bötefür ist Student der Betriebswirtschaft und Team-OS/2-Mitglied seit 1994. Als Cerified Engineer hält er Händler- und Kundenschulungen.

Herwig Diessner, geb. 1972 in Aalen, ist seit 1992 bei der IBM angestellt und hat sich von Anfang an mit OS/2 beschäftigt. Seit der Vorstellung von VoiceType war er bis 1995 im Marketing und leistete 2nd Level Support für VTD (auch Händlerschulungen). 1996 verbrachte er einige Zeit im Labor der IBM in Austin/Texas für die Vorbereitung der Vorstellung von Warp 4. Herwig Diessner ist seit 1995 im Vertrieb für die Automobilindustrie. Im Team OS/2 ist er seit November 1994 Mitglied.

Wolfgang Engel programmiert OS/2 seit dem Erscheinen der Version 2.0. Nach dem abgeschlossenem Studium der Rechtswissenschaften verlegte er sich auf die EDV. Er ist im Bereich Electronic Publishing der Hüthig Fachverlage GmbH tätig und freut sich über Anregungen und Kritik unter CIS 101527,3124.

Michael Frieß arbeitete nach seinem Informatikstudium 1989 als OS/2-Entwickler im IBM-Entwicklungslabor Böblingen. Er übernahm 1992 die Konzeption und Erstellung eines bereichsweiten LANs und von plattformübergreifenden Entwicklungsumgebungen. Er wechselte 1994 zur IBM Informationssysteme GmbH und ist seither im Software-Support für Entwicklungsprodukte und Objekttechnologien tätig. Seine Arbeitsgebiete sind hierbei VisualAge for C++ und OS/2-Portabilität.

Hendrik H. Fulda arbeitet mit OS/2 seit den ersten Beta-Versionen der Version 2.0 und ist einer der ersten IBM Certified OS/2 Engineers Deutschlands. Er ist der Initiator der WWW-Seiten des Team OS/2 Deutschland und arbeitet seit 1995 als IT-Berater und Dozent im Bereich Netzwerke und OS/2.

Norbert Heller ist seit 1995 im Team, studiert Betriebswirtschaft und hat schon viele Shareware-Programme entwickelt.

Ina Herbert ist freie Journalistin und schreibt für verschiedene Buch- und Zeitschriftenverlage. Sie arbeitet mit OS/2 seit der Version 1.0.

Thorsten Kowalski ist seit 1993 Team-Mitglied. Er ist Betriebswirt und arbeitet als kaufmännischer Angestellter. Seine Hobbies sind der PC, OS/2 und Musik. In seiner Eigenschaft als Certified Engineer hält er auch viele Schulungen und macht Vorführungen von OS/2.

Raimund Mann hat bereits Erfahrungen als Buchautor und arbeitet bei Lotus im Support. Ab Ende 1996 wird er in seinem neuen Betätigungsfeld Lotus Notes tätig sein.

Oliver Mark, Diplom-Betriebswirt für Wirtschaftsinformatik, studierte Wirtschaftsinformatik an der Berufsakademie in Mannheim. Seit 1990 ist er bei der IBM im Service-Bereich tätig. Aufgabenschwerpunkte sind Kundenprojekte im Bereich Systems Management (Softwareverteilung, Netzwerkmanagement, LAN Server Verwaltung). Er ist im Team seit den Anfangstagen und ist als regionaler Vertreter für den Raum Rhein/Neckar zuständig. Dort leitet er auch eine OS/2 User Group. Viele Händlertage und Messen mit der IBM liegen bereits hinter ihm.

Florian Piekert ist Student der Informatik und Team-Mitglied seit 1993. Als echter ftp-Freak kennt er das Internet aus- und inwendig. Auch er ist Certified OS/2 Engineer.

Andreas Portele ist Team-Mitglied und hat schon viel sehr bekannte Software für die Multimedia-Unterstützung von OS/2 entwickelt, unter anderem stammen von ihm einige interessante Multimedia-CODECS.

Christoph Pürckhauer arbeitet bei der IBM als Leiter für Objekttechnologien. Er hat schon deswegen viel Know-how zu OpenDoc.

Bernd Rohrbach studiert an der Fachhochschule Darmstadt Automatisierungstechnik mit Schwerpunkt Datentechnik. Seit der Einführung von OS/2 2.0 beschäftigt er sich mit OS/2, DB/2 und Netzwerk-Kommunikation. Bernd Rohrbach ist Certified Engineer und trat Anfang 1995 dem Team bei. Er ist der Herausgeber dieses Buchs.

Anthony Rudd hat bereits mehrere englischsprachige Bücher bei Wiley und Springer zu Rexx veröffentlicht. Er arbeitet in Nürnberg und ist ein excellenter Rexx-Kenner auf allen Plattformen.

Uwe Schlenther ist im Bereich Software-Entwicklung, EDV-Beratung und Schulung selbständig tätig. Er benutzt OS/2 seit Version 2.0, erwarb 1996 den Certified OS/2 Engineer und ist Mitglied im Team OS/2 Deutschland seit dessen Gründung. Der Schwerpunkt seines Mathematikstudiums an der Universität Stuttgart lag im Bereich der konstruktiven Differentialgeometrie.

Jürgen Sting ist gelernter Bankkaufmann und hat sich bereits in seinem 9. Lebensjahr einen eigenen PC gekauft. Er benutzt OS/2 seit der Version 2.0 und war einer der ersten Team-Mitglieder. Auch er ist Certified OS/2 Engineer.

Dimitrios Tsoukalas ist Student der Informatik und Team-Mitglied seit 1994. Er ist Certified OS/2 Engineer, stellt OS/2 auf Messen vor und hält Schulungen. Er kennt OS/2 seit der Version 2.0.

Harald Wilhelm hat nach seiner Ausbildung zum Industriekaufmann und Versicherungskaufmann den Umstieg zur EDV vollzogen. Mit den erworbenen EDV-Kenntnissen während einer Ausbildung zum Organisationsprogrammierer machte er sich 1991 selbständig. Er nahm entscheidend Teil an mehreren Entwicklungsprojekten deutscher und europäischer Großunternehmen und schreibt für diverse EDV-Fachzeitschriften. Seine Stärken liegen in der Entwicklung von Datenbankanwendungen unter dem OS/2 Presentation Manager.

1 Die Grundinstallation von OS/2 Warp V4

von Oliver Bötefür

1.1 Vorab

Heutige Betriebssysteme arbeiten mit Installationstools oder -Routinen. Doch was ist, wenn diese Routinen »versagen«?

Eventuell haben Sie Hardware, die nicht direkt vom Betriebssystem unterstützt wird. Schnell hört der Spaß am neuen Betriebssystem auf. Geld und Zeit ist vielleicht umsonst investiert worden und Sie wenden sich wieder Ihrem alten System zu und können die neuen, zum Teil aufregenden Möglichkeiten des neuen Systems nicht nutzen.

Auch OS/2 Warp Version 4.0 hat so ein Installationsprogramm und leider hat auch dieses ein paar Klippen, die, allerdings mit etwas Wissen, sehr einfach zu umschiffen sind. Dieses Kapitel hat deshalb nicht zum Ziel, die einzelnen Menüs der Installation zu beschreiben, sondern Ihnen Hilfen an die Hand zu geben, die Sie benötigen, falls Sie an solche Klippen kommen und sie umschiffen müssen.

Deshalb werden Sie in diesem Kapitel auch keine Screenshots oder Grafiken finden. Ich will Ihnen die Möglichkeit geben, mit Bildschirm und Buch zusammen OS/2 zu erobern und auf Ihrem Rechner zu installieren.

Bevor Sie ein neues Betriebssystem installieren, sollten Sie sich einige Gedanken machen, wie Ihr System hinterher aussehen soll, welchen Zweck Sie mit der Neuinstallation verfolgen und wie Sie hinterher mit dem System arbeiten wollen. Das gilt auch für OS/2.

Soll OS/2 über eine bestehende Version installiert werden? Sollen mehrere Betriebssysteme auf Ihrem Rechner laufen? Oder haben Sie ein bestehendes DOS/Windows laufen und OS/2 soll parallel dazu laufen?

Alle diese Fragen sollten Sie sich vor der Installation stellen, da sie den Ablauf der Installation beeinflussen und zu Datenverlust führen können, wenn Sie sich nicht rechtzeitig Gedanken machen.

OS/2 wird seit der Version Warp Connect nur noch als CD-Version ausgeliefert und ab der Version 4.0 nur noch mit integriertem Windows. Die Frage »Welche Version habe ich denn nun?«, entfällt also.

1.2 Überlegungen vor der Installation

Um die Installation möglichst sicher durchführen zu können, brauchen Sie Klarheit über den Umfang an Plattenplatz, den das Betriebssystem beansprucht.

Sie sollten darauf achten, daß OS/2 nicht auf komprimierte Platten installiert werden kann und sich auch überlegen, wie das System hinterher aufgebaut sein soll.

Sollen alle Programme und Daten auf einer Partition liegen oder voneinander getrennt sein?

Planen Sie, verschiedene Betriebssysteme auf Ihrem Rechner zu installieren, muß vorher der Boot-Manager installiert werden. Auch welche Dateiverwaltung sinnvoll ist, muß vor der Installation bedacht werden.

So, nun raucht erst einmal der Kopf. Aber Sie werden sehen, so schwer ist dieses alles gar nicht.

Eine wichtige Frage sollten Sie sich allerdings vor der Installation stellen:

Reicht meine Hardware, wird OS/2 überhaupt auf meinem System laufen?

OS/2 braucht als Minimalvoraussetzung einen 486DX-33MHz-Prozessor und mindestens 12 Mbyte Arbeitsspeicher. Auf der Festplatte sollten mindestens 150 Mbyte nur für das System frei sein.

Wollen Sie mit VoiceType, der Spracherkennung, arbeiten, erhöht sich der Minimalbedarf auf einen Pentium 75 MHz oder äquivalent, 24 Mbyte Arbeitspeicher, eine über 200 Mbyte große Platte, eine 16-Bit-Soundkarte und ein von OS/2 unterstütztes Mikrofon.

Empfehlenswert ist natürlich ein Modem, gängige Netzwerkkarten werden unterstützt.

Weitere Überlegungen

Haben Sie ein DOS-/Windows-System und soll dieses erhalten bleiben, brauchen Sie auch hier nur einfach OS/2 zu installieren, den Rest macht OS/2 automatisch, das heißt, nach der Installation haben Sie die Möglichkeit, sowohl OS/2 als auch DOS im sogenannten Dual-Boot zu starten. Dual-Boot hat nur einen entscheidenden Nachteil, das System wird beim Booten des DOS nicht ordentlich heruntergefahren, so daß bei jedem OS/2-Neustart ein ChkDsk gefahren wird, was sehr zeitaufwendig ist, vor allem bei großen Platten. Die Dateien »config.sys« und »autoexec.bat« müssen dauernd hin und her kopiert werden, was die Systemsicherheit nicht gerade fördert.

Außerdem ist Dual-Boot heute nur noch in den seltensten Fällen nötig, da die DOS- und Windows-Sessions unter OS/2 meist stabiler laufen als die Originalversionen und über die »Settings«, die sicher so manchen verwirren mögen, über eine sehr hohe Steuerbarkeit und Lauffähigkeit der Sessions verfügen.

Sollen, wie vorgeschlagen, mehrere Partitionen genutzt werden, weil Sie verschiedene Betriebssysteme installieren wollen oder Sie eine strukturierte Plattenarchitektur aufbauen wollen, muß erst partitioniert werden.

Dazu müssen Sie, wie bei anderen Betriebssystemen auch, das Programm FDisk aufrufen und die Platten entsprechend einrichten.

Sinnvoll sind: Eine Systempartition, die mindestens 150 Mbyte groß ist, eine Programmpartition, auf die Sie alle Programme installieren und eine Datenpartition. Die Trennung empfiehlt sich aus der Praxis; wenn System, Programme und Daten auf getrennten Partitionen vorliegen, erleichtern sie das Sichern der Daten, unterstützen das Installieren von Programmen und falls der Fall doch einmal eintritt, daß das Betriebssystem gestört oder sogar zerstört wird, die Neuinstallation.

Diese Dreiteilung hat den entscheidenden Vorteil, daß nur noch die Datenpartition archiviert werden muß, denn die Programme kann man sich relativ einfach wieder installieren. Sollte das System tatsächlich neu installiert werden müssen, bleiben alle Programme und Daten erhalten.

Allerdings installieren immer noch viele Programme DLL-Dateien und andere Daten in die Systemverzeichnisse, das bedeutet, daß einige Programme doch neu installiert werden müssen.

Legen Sie sich noch einen Pfad für die Konfigurationsdateien des Systems und der Programme, ersparen Sie sich bei Neuinstallationen sogar den Aufwand der Neukonfiguration.

Auch die Archivierung der Arbeitsoberfläche von OS/2 kann auf diese Partition gelegt werden, indem Sie in dem Kontextmenü der Arbeitsoberfläche in den Einstellungen, den Pfad auf diese Partition legen.

Hardwarekonflikte

Die Hardwaresteuerung erfolgt über Adressen, Interrupts und DMA-Kanäle. Diese werden vom BIOS vorgegeben und müssen beachtet werden. So auch von OS/2. Leider greifen aber manchmal Karten auf Adressen zu, die schon von anderen Karten belegt sind. Und wir haben einen Interessen- besser Adressenkonflikt.

Um einen Überblick zu haben, sollten Sie sich, vor allem bei Nachinstallationen von Hardware, mit dem Systemtool RMVIEW anzeigen lassen, wie Ihre Hardware belegt ist und welche Adressen noch frei sind.

Leider führen diese Adressenkonflikte schon manchmal bei der Installation zu Problemen. Ich will hier nicht auf die einzelnen Adressierungen eingehen, weil es darüber schon genügend Literatur gibt, aber darauf hinweisen, sich genau über die eingebaute Hardware zu informieren und vielleicht sich selber eine Übersicht zu erstellen, wo was wie in Ihrem System belegt ist, um Probleme zu vermeiden. Meist können Sie, indem Sie einen Jumper auf der Karte umsetzen oder eine karteneigene Software starten, die Adressen ändert und so Probleme vermeidet. Später, nach der Installation, steuern Sie dann Ihre Hardware durch OS/2 mit der »config.sys« (siehe auch dort).

Während der Installation

Fangen wir doch einfach mal an:

OS/2 soll auf Ihre Festplatte, die noch »jungfräulich« und größer als 150 Mbyte ist und nur mit OS/2 arbeiten soll, installiert werden.

Sie nehmen Ihr OS/2-Paket, stecken die beiliegenden drei Disketten nacheinander in das Laufwerk und ab geht die Post.

Ist Ihre Hardware nicht defekt, sind alle Karten sauber konfiguriert und in Ihrem Rechner steckt keine ungewöhnliche Hardware (Hardware, die keine OS/2-Unterstützung hat). Dann steht der Installation eigentlich nichts mehr im Wege. Leider wird immer noch Hardware ohne entsprechende Treiber entwickelt. IBM kann nicht laufend für alle Geräte Treiber entwickeln, sonst gäbe es bald kein neues OS/2 mehr.

Empfehlen möchte ich Ihnen grundsätzlich, nicht die *Standardvariante* der Installation, sondern die *Benutzerdefinierte Installation*, da Sie hier die Möglichkeit haben, Fehler zu korrigieren und in das Installationsgeschehen einzugreifen.

Tip: Sollten Sie bei der *Benutzerdefinierten Installation* Schwierigkeiten bekommen, ist es manchmal sinnvoll, die *Standardinstallation* auszuprobieren und danach die Komponenten zu ändern. So kann häufig defekte oder im Layout veränderte Hardware aufgespürt werden.

Beim Start der Installation von OS/2 können einige Schwierigkeiten auftreten. Zum einen werden bei der Installation von OS/2 nicht alle Controller unterstützt, obwohl es schon Treiber dafür gibt. Teils weil die Hardware noch neu ist, teils weil der Hersteller seine Treiber IBM nicht rechtzeitig zur Verfügung gestellt hat.

Andererseits gibt es CD-ROM-Laufwerke, die von OS/2 nicht gefunden werden oder Grafikkarten, für die Sie unbedingt die beiliegenden Treiber nutzen müssen.

Solche Probleme können einem die Installation von OS/2 schon etwas vermiesen, als Trost bleibt nur, daß bei anderen Betriebssystemen die gleichen Probleme gibt, nur daß es bei OS/2 Auswege aus der Misere gibt.

Wenn Sie während des Teils des Installationsprozesses, der von Disketten ausgeführt wird, ein Problem feststellen, legen Sie die Installationsdiskette ein, und starten Sie Ihr System erneut. Gehen Sie anschließend – wenn der folgende Bildschirm erscheint – wie folgt, vor:

WIEDERHERSTELLUNGSOPTIONEN

ESC	Der Systemstart wird mit der ungeänderten Datei »\config.sys« fortgesetzt.
F2	Eine Befehlszeile wird aufgerufen (keine Dateien werden ersetzt, die urspruengliche Datei »config.sys« wird verwendet).
F5	Die vollstaendige Hardwareerkennung wird aktiviert.
F6	Die Hardwareerkennung wird inaktiviert.

Drücken Sie in diesem Auswahlfenster die Taste [F6]. Sie inaktivieren dadurch die automatische Erkennung Ihrer Hardware.

Bleibt das System weiter stehen, können Sie wie folgt, die gerade geladenen Einheitentreiber anzeigen lassen:

1. Legen Sie die Installationsdiskette ein, und starten Sie Ihr System erneut.
2. Drücken Sie die Tasten [Alt]+[F2], wenn in der linken oberen Ecke der Anzeige das weiße Rechteck und »OS/2« erscheint. Dann zeigt Ihnen OS/2 die Einheitentreiber an, die gerade geladen werden (»Loading xxxx«). Bleibt das System nach einem Treiber stehen, so ist das Problem meist bei dem zuletzt angezeigten Treiber zu suchen.
3. Editieren Sie die Datei »config.sys«, und kommentieren Sie die Anweisung für diesen Einheitentreiber, der nicht geladen werden kann, aus.
4. Starten Sie Ihr System erneut.

Leider sind beide Optionen nicht gleichzeitig möglich. Aber diese Option zeigt Ihnen, wo OS/2 bei Ihrer Installation auf Schwierigkeiten stößt, da es an diesem Treiber stehen bleibt.

Wenn Sie übrigens bei einem System mit 2,88-Mbyte-Diskettenlaufwerk Probleme haben, weil das Diskettenlaufwerk nicht erkannt wird: Schalten Sie im BIOS während der Installation auf 1,44 Mbyte um. Das Problem sollte dann behoben sein,

1.3 Der Boot-Manager

Der Boot-Manager ist ein Bootmenü, das dem normalen Bootvorgang übergeordnet ist und somit auch eine eigene primäre Partition. Er ermöglicht es, verschiedene Betriebsysteme oder Systemvarianten auf einem Rechner zu installieren und je nach Bedarf zu starten. Deshalb ist es sinnvoll, den Boot-Manager immer zu installieren, auch wenn Sie ihn derzeit nicht benötigen. Aber wer weiß, was die Zukunft bringt!

Der Boot-Manager braucht normalerweise 1 Mbyte, allerdings bei größen Platten mehr Platz auf der Festplatte. Bei einer 3-Gbyte-Platte können es z.B. 7 Mbyte werden. Er enthält Angaben über die auf der Festplatte befindlichen Betriebssysteme und zeigt diese bei dem Bootvorgang an, wenn sie vorher im Boot-Manager initialisiert worden sind.

Installieren können Sie den Boot-Manager nur unter der Option *Benutzerdefinierte Installation*. Ein weiterer Grund, weshalb Sie diese Option wählen sollten und nicht die Standardinstallation.

Nach der Auswahl der benutzerdefinierten Installation haben Sie die Möglichkeit, FDisk mit dem Punkt *Anderes Laufwerk angeben* aufzurufen. Dort partitionieren Sie Ihre Platte oder legen OS/2 auf eine andere Partition als C:, falls Sie dieses wünschen.

Allerdings: Mehr als fünf verschiedene System kann der Boot-Manager nicht aufnehmen. Dies liegt nicht an der Software, sondern an der Hardware.

Unter FDisk erkennen Sie auch die vorhandenen Platten und Partitionen.

Richten Sie nun den Boot-Manager mit der Option *Boot-Manager installieren* ein. Sie werden nun gefragt, ob der Manager am Anfang oder am Ende des freien Festplattenbereichs installiert werden soll. Da einige Betriebssysteme unbedingt den Bootsektor am Anfang der Platte brauchen, empfiehlt es sich, den Boot-Manager immer an das Ende des freien Bereichs zu legen. Die gilt allerdings nicht für Festplatten mit mehr als 1024 Zylindern. Auch hier liegt es nicht an der Software, sondern am System-BIOS.

Sobald der Boot-Manager installiert ist, müssen Sie ihn noch als *Startpartion* definieren.

Nun haben Sie die Möglichkeit, die anderen Partitionen mit dem Menüpunkt *Partition erstellen* festzulegen. Sie können drei primäre Partitionen anlegen. Allerdings ist nach dem Booten nur eine der drei Partitionen sichtbar.

Eine Partition brauchen Sie für OS/2 Warp. OS/2 muß aber nicht auf einer primären Partition angelegt werden, sondern kann auch in einer erweiterten Partition oder auf einem anderen Laufwerk liegen.

Nach dem Anlegen der Partition drücken Sie bitte noch einmal die Eingabetaste für diese Partition. Wählen Sie bitte in dem jetzt erscheinenden Menü die Option *Installationspartition...*, damit der Boot-Manager nach dem Bootvorgang weiß, daß Sie OS/2 hierhin installieren wollen. Nun können Sie alle anderen Partitionen bestimmen.

Mit dem Menüpunkt *In Menü aufnehmen...* geben Sie bitte die Namen der einzelnen Partitionen ein und initialisieren Sie die Möglichkeit, diese Systeme später zu starten.

Jetzt fehlt nur noch die Konfiguration des Boot-Managers selber.

Gehen Sie mit den Pfeiltasten auf die Partition, die Sie standardmäßig laden wollen, drücken dann die Eingabetaste.Sie erhalten jetzt die Möglichkeit, hier die Startwerte festzulegen:

1. Welches System immer auf jeden Fall gestartet werden soll, falls Sie keine Auswahl treffen.

2. Nach welcher Zeitspanne der Boot-Manager starten soll, falls Sie nicht die Eingabetaste betätigen.

Sie haben jetzt noch die Möglichkeit, den Modus zu ändern. Beim »Standardmodus« wird nur der Name der Partition angezeigt, beim »Erweiterten Modus« noch zusätzlich Partitionstyp, Dateisystem und Zugriffsmöglichkeit.

Wenn Sie nun FDisk mit [F3] beenden, sind Ihre Einstellungen gesichert und Sie können mit der Installation fortfahren.

Die Installation des Boot-Managers kann auch noch nachträglich erfolgen.

Die Vorgehensweise ist dieselbe, außer daß Sie den Manager nur an das Ende einer Platte legen können und Sie so, falls Sie nicht vorher partitioniert hatten, eventuell Daten verlieren können, da Sie erst Platz auf der Platte schaffen müssen. Moderne Tools wie »Partition

Magic« von PowerQuest Corp. umgehen diesen Datenverlust, da sie in der Lage sind, während des laufenden Betriebs, per Schiebeschalter, die Partitionsgrößen zu verändern.

Haben Sie einen Fehler, entweder in FDisk oder beim Installieren des Boot-Managers gemacht, verlassen Sie einfach FDisk, ohne die Änderungen zu speichern.

Stellen Sie im laufenden Betrieb fest, daß Fehler im Boot-Manager auftreten, können Sie diese beheben, ohne daß Ihre Daten verloren gehen. Der Boot-Manager kann mit dem Befehl *SET-BOOT* auch nachträglich gesteuert werden. Rufen Sie dazu von der Befehlszeile »help SET-BOOT« auf, dann erfahren Sie mehr zu diesem Befehl und können ihn dann einsetzen.



Sollte der Boot-Manager nach einer Installation eines anderen Betriebssystems nicht mehr funktionieren, rufen Sie bitte einfach FDisk (auch DOS-FDisk) auf, setzen Sie dann die BootManager-Partition wieder auf aktiv und der Boot-Manager arbeitet wieder wie Sie es gewohnt sind.

Abb. 1.1: Im FDisk von DOS ist der Boot-Manager als NON-DOS-Partition vermerkt. Setzen Sie ihn nach der Installation eines anderen Systems wieder auf **a**ktiv.

1.4 Einheitentreiber

Grundlagen

Sie können bei der Installation von Hardware, die nicht direkt von OS/2 unterstützt wird, diese trotzdem einfügen.

Fangen wir damit an, daß Ihr Controller von OS/2 nicht unterstützt wird. Sie haben aber OS/2-Treiber mitgeliefert bekommen.

Löschen Sie auf der Installations-Diskette 1 die Treiber, die Sie für Ihr System nicht brauchen (Endung ».add«) und auch die dazugehörenden »Snopper« (Endung ».snp«). Nun haben Sie Platz für Ihren Treiber. Geben Sie in der »config.sys« der Diskette 1 Ihren Treiber an und löschen Sie dort ebenfalls die nicht benötigten Treiber. Fügen Sie zu Ihrem Treiber die in der Hardwaredokumentation angegeben Parameter an und speichern Sie nun die Datei auf die Diskette.

Wenn Sie jetzt von Disketten booten, drücken Sie bitte, wenn oben links auf Ihrem Bildschirm der weiße Block erscheint, die Tastenkombination [Alt]+[F1]. Wählen Sie, wie beschrieben, bitte [F6], um die Hardwareerkennung abzustellen.

Jetzt haben Sie zwei Möglichkeiten:

1. Sie drücken, bevor Sie die Diskette 2 aus dem Laufwerk nehmen sollen, die Taste [F3] und kopieren die Treiber von Hand auf die Festplatte in das Verzeichnis »\os2\boot\«

oder Sie fügen

2. in der »config.sys« den Befehl *set copyfromfloppy=1* ein.

Bei der zweiten Möglichkeit liest OS/2 nicht die Disketten von der CD ein, sondern Ihre Bootdisketten.

Wird Ihr CD-ROM nicht unterstützt, gilt im Prinzip dasselbe wie für den Controller. Allerdings müssen Sie darauf achten, daß in der »config.sys« das Statement *device=\OS2cdrom.dmd* enthalten ist und auch der entsprechende Treiber »cdrom.flt« eingetragen werden muß.

Installation der Grafikkarte

Sie haben eine Grafikkarte, die eigene OS/2-Treiber braucht. Auch dies ist kein Drama. Geben Sie bitte bei der Installation an, daß OS/2 nur eine VGA-Auflösung installieren soll. Einige Karten, wie beispielsweise die der Firmen ELSA und Spea, haben ein eigenes Installationsprogramm, mit dem der Kartentreiber installiert wird, andere, wie Grafikkarten der Firma ATI, müssen später über das Objekt *Einheitentreiber installieren* nachinstalliert werden.

Sollte OS/2 Ihren Grafikchip erkannt haben, der Hersteller aber wieder einmal das Layout verändert haben und Ihr Bildschirm flimmert oder bleibt nach der Installation schwarz, haben Sie über das [Alt]+[F1]-Menü die Möglichkeit, den Bildschirmtreiber auf VGA zurückzusetzen, indem Sie nach der kompletten Installation rebooten und dann im Menü auf [F3] drücken.

Mit [F4] können Sie von diesem Schirm aus erneut in das Installationsprogramm kommen, um eventuelle Veränderungen der Hardware, vor einem kompletten Hochfahren des Systems, nachzuinstallieren.

NE2000-kompatible Netzwerkkarten

Schwierigkeiten kann es noch einmal bei NE2000-kompatiblen Netzwerkkarten geben. OS/2 unterstützt diese Karten nur, wenn sie »Shared Memory« benutzen. Viele der Kompatiblen unterstützen aber kein Shared Memory. Hier hilft nur entweder vor der Installation sich einen passenden Treiber vom Hobbes-Server zu laden und während der Netzwerkinstallation über *Anderer Adapter* den Treiber einzuspielen oder aber die Netzkarte erst einmal zu entfernen, OS/2 komplett zu installieren und dann die Karte zu konfigurieren, wie es in den Kapiteln über TCP/IP und LAN beschrieben ist.

Falls die Statusanzeige während der Installation stoppt

Sollte der Statusanzeiger während der Installation sich nicht mehr weiterbewegen und sowohl die Betriebsanzeigen des CD-ROM- und Festplattenlaufwerks keine Installationsaktivität erkennen lassen, gehen Sie wie folgt vor:

Verkleinern Sie die Statusanzeige auf Symbolgröße, indem Sie die Tastenkombination [Alt]+[Shift]+[F3] drücken. Daraufhin erscheint der Knopf für »Symbolgröße« in der oberen rechten Ecke der Anzeige. Danach klicken Sie den Knopf für »Symbolgröße« an. Sie können nun ein OS/2-Fenster öffnen, um die folgenden Dateien zu prüfen, da bei der Installation jedes OS/2 Warp-Netzwerkprodukts Protokolldateien erstellt werden. Diese Protokolldateien werden in den Unterverzeichnissen des Verzeichnis »\ibminst\logs\« gespeichert:

– In der Datei »\ibminst\logs\locinstu\local.l2« wird der Installationsstatus des OS/2 Warp-Netzwerkprodukts mit der Angaben zu den installierten Produkten protokolliert.

– Im Verzeichnis »\ibminst\rsp\local\« stehen die Antwortdateien für die lokale Installation von OS/2-Netzwerkprodukten.

– Im Verzeichnis »\ibminst\rsp\remote« finden Sie die Antwortdateien für die Ferninstallation von OS/2-Netzwerkprodukten.

1.5 Formatieren während der Installation

Es gibt während der Installation von OS/2 mehrere Möglichkeiten, die Festplatte zu formatieren.

Die erste Möglichkeit ist über die *Benutzerdefinierte Installation* gegeben. Allerdings dürfen Sie sich nicht wundern, wenn das Format sehr schnell ist. Es ist ein Quickformat. Es werden also nur die Verzeichniseinträge gelöscht und nicht die Platte komplett formatiert. Um nun ein Komplettformat zu erreichen, müssen Sie nach dem Durchlaufen der ersten drei Disketten mit [F3] die Installation verlassen und auf der Kommandozeile den Befehl Format mit Parametern aufrufen. Wenn Sie nach einem gründlichen Format mit HPFS arbeiten wollen, lautet der Befehl:

```
FORMAT x: /FS: HPFS /L (long)
```

Jetzt ist die Platte sauber formatiert, selbst wenn Sie später ChkDsk mit dem Parameter »/f« aufrufen, werden keine alten Dateien mehr restauriert und Sie erhalten so auch keinen Datensalat. Sollten Sie dennoch mal ein Quickformat für FAT benötigen, lautet der Befehl:

```
FORMAT x: /FS:FAT /Q (quick)
```

Die Parameter sind gegeneinander austauschbar, mehr erfahren Sie, wenn Sie von der Kommandozeile »help format« aufrufen oder in der Referenzhilfe von OS/2 nachschauen.

Eine weitere Möglichkeit, während der Installation zu formatieren, haben Sie entweder, indem Sie mit dem obengenannten Befehl [Alt]+[Shift]+[F3] das Installationsmenü verkleinern und

von dort über die Ordner formatieren oder aber auch im Installationsprogramm selber. Im dritten Bildschirm finden Sie eine Ordnerleiste mit den Punkten *Optionen*, *Softwarekonfiguration* und *Hilfe*. Bei *Optionen* können Sie erstens *Installieren* lassen, als wenn Sie unten auf *weiter* klicken würden. Zweitens können Sie unter *Formatieren* Formatbefehle eingeben. Unter dem dritten Punkt *Befehlszeilen* besteht die Möglichkeit, OS/2-Befehlszeilen aufzurufen.

Zum Thema »Formatieren« möchte ich Ihnen noch einen wichtigen Hinweis geben: Wenn Sie sowohl mit FAT- als auch mit HPFS-formatierten Platten arbeiten, sollten Sie die FAT-Partitionen vor die HPFS-Partitionen legen. Der Sinn liegt darin, daß DOS HPFS-Formationen nur mit Hilfe von Zusatztools wie dem Programm »AMOS« erkennen kann und sonst, wenn von DOS gebootet wird, die Festplattenzuordnung durcheinander gerät. Dies würde bedeuten, daß viele Programme nicht mehr laufen, da sie ihre Konfigurationsdaten nicht mehr finden.

1.6 Parameter für die »config.sys«

Das gleiche Menü, mit dem Sie formatieren können, enthält auch den Punkt *Softwarekonfiguration*. Hier besteht die Möglichkeit, schon bei der Installation Einträge in der »config.sys« zu ändern. Schauen Sie sich diesen Punkt einmal in Ruhe an und vergleichen ihn mit den Einträgen, die im Kapitel über die »config.sys« erklärt werden.

Wenn Sie die Parameter für Ihre Soundkarte verändern müssen, haben Sie dazu die Möglichkeit, indem Sie auf dem ersten Installationsbildschirm auf das Symbol für die Soundkartenauswahl klicken, dort Ihre Soundkarte auswählen und – falls sie noch nicht erkannt wurde – die Karte dem System hinzufügen. Danach haben Sie die Möglichkeit, den Punkt *Einstellungen* anzuklicken, um dort den DMA-Kanal und die Adressen einzustellen.

1.7 OS/2-Installation von der Festplatte

OS/2 muß nicht von CD installiert werden. Das System ist darauf eingerichtet, von jedem beliebigen Speichermedium installiert zu werden. Sie können also auch von Ihrer Platte aus eine OS/2-Installation durchführen. Dies kann nötig sein, wenn Ihr CD-ROM streikt und Sie nur mit dem DOS-Treiber an Ihr System kommen.

Kopieren Sie bitte die vollständigen Verzeichnisse »cid«, »os2image« und »ibminst« auf Ihre Festplatte. Außerdem muß die Datei »os2se20.src« in das Rootverzeichnis der Platte gespielt werden. Günstiger ist es natürlich, eine eigene Partition für diese Daten zu haben, da Sie bei der Installation nicht Formatieren dürfen. Die Daten sind sonst weg! Jetzt müssen Sie nur noch mit dem Befehl »makedsks.bat« die ersten drei Disketten erstellen lassen und dann die »config.sys« auf der zweiten Diskette anpassen. Dazu fügen Sie bitte nach dem Befehl *set path* den Pfad »os2image« ein und das neue Statement *set sourcepath=x:\os2image* hinzu.

Wenn Sie jetzt mit den ersten drei Disketten booten, sollte OS/2 automatisch die Daten auf Ihrer Platte finden und von dort eine Installation durchführen. Falls das Installationsprogramm streiken sollte oder Sie ein pures OS/2 Warp ohne Netzunterstützung (LAN) laden wollen, brauchen Sie den erwähnten Befehl »makedsks.bat« erneut. Erstellen Sie wieder die ersten drei Disketten. Fügen Sie anschließend die Treiber für CD-ROM und Controller in die »config.sys« ein, booten Sie und beenden Sie das Installationsprogramm mit [F3]. Jetzt wechseln Sie bitte auf Ihre CD und rufen Sie von dort den Befehl »install.cmd« auf.

Diese Installationsroutinen ist noch mächtiger als das Installationsprogramm, das von OS/2 automatisch aufgerufen wird.

Sie brauchen die »install.cmd« auch dann, wenn Sie kein Netz installiert haben, aber das »IAK« (siehe auch dort) laden wollen. Sie haben durch die Routine die Möglichkeit, gezielt die Internetunterstützung zu laden, die Sie für Ihre Bedürfnisse benötigen.

1.8 CID

Wie bereits erwähnt: OS/2 muß nicht von CD installiert werden. Im Zuge der Netzwerke und da OS/2 ein netzwerkunterstützendes Betriebssystem ist, kann OS/2 auch aus einem Netzwerk heraus installiert werden. Zum einen gibt es die Möglichkeit, nachdem die Installationsdisketten an Ihr Netzwerk angepaßt worden sind, das System von Hand zu installieren. Aber IBM hat noch eine automatisierte Methode eingeführt. Diese nennt sich CID.

CID steht für:

– **C**ONFIGURATION

– **I**NSTALLATION

– **D**ISTRIBUTION

Bei CID gibt es verschiedene Varianten. Sie können tatsächlich die gesamte Installation von einem Server aus als eine Ferninstallation, durchführen, Sie können sich die Daten von Ihrem Client aus aber auch aus einem Netz heraus laden.

Es gibt aber auch eine kleine Variante. Das heißt, wenn ein System einmal installiert ist, muß man bei einer Neuinstallation nicht mehr durch alle Menüs wandern, sondern kann auch hier die Installation, zumindest für das pure OS/2 Warp, automatisieren.

Diese Variante möchte ich Ihnen hier einmal vorstellen:

Als erstes müssen Sie für diese Installation eine sogenannte Response-Datei (».rsp«), also eine Antwortdatei, erzeugen. Welche Parameter Sie dazu benötigen, zeigt Ihnen OS/2 in der Datei »sample.rsp«, die entweder im Verzeichnis »\os2\« liegt, die Sie aber auch auf der beiliegenden CD-ROM im Verzeichnis »CID« finden. Nachdem Sie OS/2 installiert haben, hat das System für Sie schon eine solche Response-Datei mit dem Namen »user.rsp« erzeugt und ebenfalls im Verzeichnis »\os2\« abgelegt.

Diese Datei enthält nach der Installation alle Daten des installierten Systems. In der Datei müssen nur die Parameter aufgeführt werden, die für Ihr System nötig sind. Andere Parameter, die zwar möglich, aber nicht nötig sind, sollten in der »user.rsp« fehlen.

Passen Sie bitte nun die Installationsdisketten an die CID-Installation an:

Ändern Sie bitte in der »config.sys« das Statement

```
set os2_shell=sysinst2.exe
```

in

```
set os2_shell=rspinst.exe a:\user.rsp
```

Nun wird ein reines OS/2 Warp auf Ihre Platte kopiert. Um Multimedia zu installieren, müssen Sie nach dem Systemstart in das Verzeichnis »\mmtemp\« wechseln. Rufen von dort aus die Datei »MInstall« mit dem Parameter »/M« auf. Auch für die Multimedia-Installation gibt es eine Responsedatei. Ein Beispiel für das Aufrufen von »minstall« mit einer Antwortdatei wäre:

```
MINSTALL /M /C:MEDIA.RSP
```

Nachdem diese Vorbereitungen abgeschlossen sind, erfolgt die Installation. Dies bedeutet, daß Ihre Daten so auf die Platte installiert werden, wie Sie es vorher bestimmt haben und nicht einfach alles auf die Platte kopiert wird.

Bei einer Netzvariante kommt jetzt der Punkt »Distribution« dazu. Das heißt, Sie haben die Möglichkeit, jeden folgenden Client individuell zu konfigurieren. Die Verteilung erfolgt also nicht nach Schema F. Wenn Sie noch mehr über die Möglichkeiten der CID-Einrichtung wissen möchten, lesen Sie bitte die Datei »readme.cid« auf der Installationsdiskette oder besorgen Sie sich bitte von IBM das Redbook CID.

Weiterführende Literatur:

Redbook CID: OS/2 Installation Techniques – The CID Guide GG24-4295-01, IBM Corp.

OS/2-Inside-Hefte, AWI-Verlag

Ihr OS/2 Warp Scout, OS/2-Inside Edition, Markt & Technik, 1995

2 Die Systemarchitektur von OS/2

von Harald Wilhelm

2.1 Sessions, Prozesse und Threads

IBM OS/2 Warp 4 unterscheidet, wenn es um die Ausführung und Verwaltung von Programmen geht, zwischen Sessions, Prozessen und Threads.

Als »Thread« (Programmfaden) gilt die Einheit eines Programms, das unter OS/2 überhaupt etwas machen kann. OS/2 verwaltet auf Thread-Ebene die zur Ausführung anstehenden Programme. Jeder Thread kann weitere Threads zur Verarbeitung weiterer Aufgaben aktivieren.

»Prozeß« ist der Oberbegriff für ein Anwendungsprogramm. Jede Anwendung besteht aus mindestens einem Thread. Einem Prozeß zugeordnet sind Systemressourcen wie Dateien und Hauptspeicherbereiche. Jeder Prozeß kann wiederum weitere Prozesse zur Ausführung bringen, wobei jeder Prozeß seine eigenen Ressourcen besitzt. Alle Threads eines Prozesses haben uneingeschränkten Zugriff auf alle Ressourcen, die diesem Prozeß zugeordnet sind.

Als »Session« (Bildschirmgruppe) wird unter OS/2 Warp 4 die Einheit bezeichnet, die Zugriff auf virtuelle Ressourcen wie den Bildschirm, die Tastatur oder die Maus erhält. Eine Session kann wiederum weitere Sessions aktivieren, aber immer nur einer Session wiederum sind die virtuellen Ressourcen zugeordnet. Die Session, der die virtuellen Ressourcen zugeordnet sind, erkennt man ganz einfach. Das ist die Session, die den Prozeß beinhaltet, mit dem der Benutzer zur Zeit arbeiten kann.

2.1.1 Threads

Als Thread wird die kleinste unter IBM OS/2 Warp 4 ausführbare Programmeinheit bezeichnet. Jeder Thread hat eine eigene Priorität, die Auskunft darüber gibt, wie wichtig die Ausführung dieses Programmteils dem Entwickler des Programms und dem System erscheint. Weiterhin besitzt jeder Thread einen eigenen Stack, das ist ein eigener Speicherbereich, sowie einen eigenen Zustandsbericht über die CPU-Register. Alle diese Daten werden als Thread-Kontext bezeichnet. Diese versetzen OS/2 Warp in die Lage, im Bruchteil einer Sekunde einen Thread zur Ausführung zu bringen.

OS/2 Warp 4 unterscheidet bei Threads zwischen drei verschiedenen Zuständen:

– *running* (zur Zeit aktiviert) – manchmal auch »active«

– *suspended* (wartet auf die Aktivierung) – manchmal auch »Ready to run«

– *blocked* (verhindert) – manchmal auch »Waiting for I/O«

Der Thread-Kontext eines aktiven Threads ist zur Zeit in der CPU geladen. Dieser Thread hat uneingeschränkten Zugriff auf alle Daten, die dem Prozeß zugeordnet sind, dem dieser Thread angehört. Der Code des Threads wird im Moment von der CPU abgearbeitet. Ein laufender Thread wird mit ziemlich hoher Wahrscheinlichkeit als nächstes in den Suspended-Modus versetzt, nämlich dann, wenn ein weiterer Thread ebenfalls auf seine Ausführung wartet, oder der Thread auf Ressourcen zugreift, deren Beschaffung nicht sofort ausgeführt werden kann.

Wurde ein Thread durch einen anderen Thread in der Ausführung abgelöst, so wird er vom System in einen Schlafzustand versetzt (suspended). Der Thread könnte zur Zeit zwar etwas erledigen, im Moment ist er aber nicht an der Reihe. Wird dieser Thread wieder vom System aktiviert, so führt er seine Arbeit exakt an der Stelle wieder fort, an der er zuvor unterbrochen wurde.

Anders sieht das bei einem Thread aus, der im Moment auf die Beschaffung von Ressourcen wartet. Ihn zu aktivieren, macht erst dann wieder Sinn, wenn diese Daten auch tatsächlich zur Verfügung stehen. Stehen diese Ressourcen jedoch wieder zur Verfügung, so wird dieser zuvor blockierte Thread mit großer Wahrscheinlichkeit sofort an die Reihe kommen. Ebenfalls sicher, wenn gewollt, erhält dieser nun erwachte Thread einen Extraschub vom System. Dieser Extraschub, auch als I/O-Burst bezeichnet, soll den Thread für die lange Wartezeit entschädigen und dem Benutzer, der auf die Ausführung seiner Programme wartet, den Eindruck vermitteln, der Thread hätte nicht übermäßig lange auf die Ausführung gewartet.

Für die Aktivierung eines neuen Threads ist ein API-Aufruf mit dem Namen DosCreateThread zuständig. Die Aktivierung eines Threads läßt sich mit geringem Aufwand an Systemzeit durchführen, so daß es auf jeden Fall Sinn macht, die Nutzung von mehreren Threads immer in Betracht zu ziehen. Nur in ganz seltenen Fällen wird der Anwendungsentwickler sich direkt mit dem DosCreateThread-API beschäftigen müssen. Da üblicherweise zur Programmentwicklung Hochsprachen wie C herangezogen werden, besitzen diese meist eine eigene Variante mit dem Namen *beginthread*. Die Nutzung dieses Funktionsaufrufs ist dem *DosCreateThread* auf jeden Fall vorzuziehen, da während der Verarbeitung von *beginthread* das System der Hochsprache auf den Multithreading-Betrieb vorbereitet wird. Der *DosCreateThread* hingegen kümmert sich lediglich ausschließlich um Belange des Betriebssystems.

Grundsätzlich endet ein Thread, wenn er an das Ende seines Programmcodes gelangt. Der explizite Abbruch von Threads kann mit dem DosKillThread-API vorgenommen werden. Wurde der Thread allerdings mit Hilfe der Hochsprachenfunktion »beginthread« aktiviert, so hilft in diesem Fall nur die »endthread«-Funktion weiter.

Auf jeden Fall vermieden werden muß die gemischte Nutzung dieser Funktionalitäten. Ein mit *beginthread* gestarteter Thread sollte auf keinen Fall mit *DosKillThread* gestoppt werden, da die Hochsprache in diesem Fall nicht in der Lage ist, ihr System ordentlich aufzuräumen.

2.1.2 Prozesse

Prozesse sind in der Hierarchie eine Stufe höher angesiedelt als Threads. Jeder Prozeß besitzt mindestens einen Thread, denn nicht die Prozesse werden ausgeführt, sondern die Threads als kleinste und einzige ausführbare Einheit unter OS/2 Warp 4. Über den einen Thread hinaus, der immer existiert, kann ein Prozeß aus bis zu 4094 weiteren Threads bestehen. Diese Angabe ist allerdings ein theoretischer Wert, denn ab einer Größenordnung von ca. 200 spricht man schon von Threadpools, deren Verwaltung spezieller Vorkehrungen bedarf. Die Umgebungsvariable THREADS in der »config.sys« (ohne SET verwenden) erlaubt die Beschränkung der systemweit nutzbaren Threads. Ein Verringern dieses Parameters spart minimalen Speicherbereich. Bei Überschreitung dieses Limits können allerdings unerwartete Ergebnisse auftreten.

Wenn auf der OS/2-Oberfläche, der Workplace Shell, ein Programm gestartet wird, so handelt es sich dabei immer um einen Prozeß. Der dabei automatisch mit eingerichtete Standard-Thread (Thread 1 oder Primary Thread) wird sich zunächst mit der Abarbeitung des Programmcodes beschäftigen. Dieser kann dann erst weitere Threads zur Ausführung bringen.

Beim Start eines Prozesses werden diesem automatisch bestimmte Ressourcen zugeordnet. Darunter befindet sich unter anderem ein virtueller Adreßraum von bis zu 512 Mbyte. Bei diesem Wert handelt es sich ebenfalls um einen theoretischen Wert, denn das System zweigt von diesem Adreßraum auf jeden Fall genügend Speicher ab, in den die Systemdateien eingeblendet werden. Mit Systemdateien sind unter anderem DLLs gemeint, auf die der Prozeß zugreifen wird.

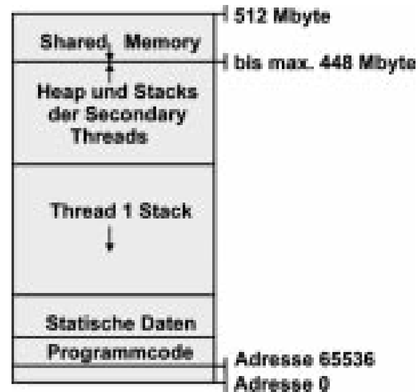


Abb. 2.1: Die Struktur eines Prozesses mit der Hauptspeicheraufteilung

OS/2 Warp 4 unterscheidet technisch zwischen dem ersten Thread eines Prozesses und allen weiteren Threads dieses Prozesses. So werden dem ersten Thread auf jeden Fall die angeforderten Hauptspeicherressourcen direkt übergeben. Dieser Hauptspeicher des ersten Threads ist sofort scharf und wird vom verfügbaren Systemspeicher abgezogen. Bei allen Hauptspeicheranforderungen der weiteren Threads erhalten diese lediglich eine Bestätigung darüber, daß noch genügend Hauptspeicher zur Verfügung steht. Das Sytem merkt sich nur in einer internen Tabelle, daß ein Thread Bedarf auf Hauptspeicher angemeldet hat, aber das System wird die-

sen Hauptspeicher nicht direkt an den Thread übergeben. Erst wenn die sogenannten Secondary Threads den Hauptspeicher tatsächlich benutzen wollen, wird ihnen dieser physikalisch zur Seite gestellt. Die Unterscheidung zwischen Primary Thread und Secondary Threads ist ein Schutzmechanismus und soll die Vergeudung von System-Hauptspeicher vermeiden helfen. Primary Threads werden vom System gestartet und zwar dann, wenn der Elternprozeß aktiviert wird. Alle Secondary Threads werden aber vom Anwendungsprogrammierer während der Programmausführung gestartet. Während das System sich über seine Handlungsweise bei der Anforderung von Hauptspeicher im klaren ist, mißtraut es den Anwendungsprogrammen zunächst und gibt ihnen den Hauptspeicher häppchenweise nur bei der tatsächlichen Nutzung.

Man muß sich das in etwa wie bei der Nahrungsaufnahme vorstellen. Hat man Hunger, dann kann die angeforderte Mahlzeit zu üppig ausfallen. OS/2 gibt den Secondary Threads nach jedem Bissen erneut eine Chance und fragt nach, ob der Secondary Thread wirklich immer noch Hunger hat.

Da alle Threads eines Prozesses unbeschränkten Zugriff auf die Ressourcen des Prozesses besitzen, muß vor der Nutzung von gemeinsamen Ressourcen der Zugriff auf diese geregelt werden. Man muß sich nur eine Speicherstelle vorstellen, die von mehreren Threads »gleichzeitig« verändert wird. Das Ergebnis wäre eher zufällig, denn immer nur die letzte Änderung bliebe aktiv. Die Synchronisation der Zugriffe auf solche gemeinsame Ressourcen wird mit Hilfe der Interprozeßkommunikation (IPC) gelöst. So bieten sich zum Beispiel Semaphoren zum Abschotten der Speicherstellen an, an denen mehrere Threads Interesse bekunden.

Ein Prozeß wird durch Aufruf des DosExecPgm-APIs gestartet. Natürlich gibt es auch hierzu die entsprechende Aufräumfunktion, das DosKillProcess-API übernimmt diese Aufgabe.

2.1.3 Sessions

In einer Session können eine oder mehrere Prozesse mit ihren jeweiligen Threads aktiv sein. Der Session, mit der der Benutzer arbeiten kann, werden die virtuellen Ressourcen wie Bildschirm, Tastatur und Maus zugeordnet. Alle Prozesse innerhalb einer aktiven Session erhalten diese virtuellen Ressourceen zugeordnet.

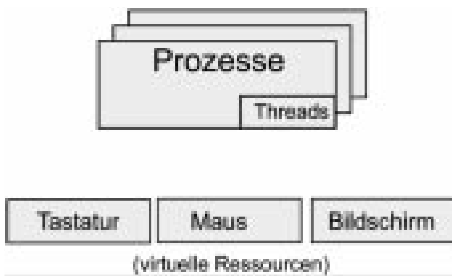


Abb. 2.2: Verschiedene Prozesse teilen sich dieselben virtuellen Ressourcen

Für die Verwaltung der Sessions ist ein spezieller Manager im OS/2 verantwortlich. Der Session-Manager unterscheidet zwischen Vordergrund- und Hintergrund-Sessions.

OS/2 Warp 4 unterscheidet weiterhin zwischen verschiedenen Typen von Sessions:

– *DOS Fullscreen Sessions*. Jede DOS-Ganzbildanwendung läuft in ihrer eigenen Session, sie kann Text und Grafik enthalten. Im System können keine oder mehrere DOS-Fullscreen-Sessions gleichzeitig aktiv sein.
– *OS/2 Fullscreen Sessions*. Alle Programme, die durch den CMD.EXE gestartet werden, erhalten ihre eigene Session. Im System können keine oder mehrere OS/2-Fullscreen-Sessions gleichzeitig aktiv sein.
– *PM Session*. Fast jedes OS/2-Programm wird in dieser zentralen Session zur Ausführung gebracht. Alle Fensterapplikationen, auch OS/2-Fenster sowie DOS-Fenster, werden von dieser einen Session abgedeckt. Es gibt nur eine einzige PM-Session pro System. Diese Session wird bereits beim Systemstart aufgebaut.
– *Detached Processes*. Alle Prozesse, die mit dem DETACH-Befehl ausgeführt werden, laufen unter der Herrschaft dieser einen Session. Es gibt nur eine einzige Detached-Session pro System. Mit Start des ersten mit DETACH aktivierten Prozesses wird diese Bildschirmgruppe zum ersten Mal aktiviert. Dieser Session werden keine virtuellen Ressourcen wie Bildschirm, Tastatur oder Maus zugeordnet, da DETACH-Prozesse diese nicht benötigen. Es handelt sich bei dieser Art von Anwendung um reine Verrichtungsprogramme, ohne jegliche Interaktion mit dem Benutzer.

2.2 Multitasking und Multithreading

Obwohl es uns bei der täglichen Arbeit anders vorkommt, sind moderne Computer mit den gestellten Aufgaben vielfach unterfordert. Dies gilt zumindest für benutzergesteuerte Aufgaben.

Als benutzergesteuert gilt die Arbeit des Benutzers mit dem Computer. Hierzu zählt die Eingabe von Texten mit der Tastatur, die Arbeit mit der Maus oder ähnlichem. Ein Benutzer kann niemals so schnell tippen, daß eine moderne CPU nicht mehr mithalten könnte. Anders sieht das bei den internen Aufgaben aus.

Eine Benutzeraktion wie ein Tastaturdruck oder ein Mausklick kann durchaus zu einer Aufgabe führen, mit der ein schneller Rechner bis an das Ende seiner Leistungsfähigkeit gebracht wird.

Der Umstand, daß ein Computer vielfach ein unausgelastetes Dasein fristet, führte zu der Idee, den Computer weitere Arbeiten verrichten zu lassen, während der Benutzer mit der Ausführung des nächsten Arbeitsschritts beschäftigt ist. Hiermit war das Multitasking geboren.

2.2.1 Multitasking

Als Multitasking wird die Verteilung der Rechenleistung eines Computers auf mehrere Aufgaben bezeichnet, die dann »gleichzeitig« durchgeführt werden. Daß diese Aufgaben gleichzeitig durchgeführt werden, stimmt selbstverständlich nicht ganz, denn eine einzelne CPU kann zur Zeit eben doch nur eine einzige Aufgabe des Computers zu einer gegebenen Zeit bewältigen. Moderne CPUs sind zwar durchaus in der Lage, intern, das bedeutet im Chip, mehrere Aufgaben gleichzeitig durchzuführen, diese Chip-internen Aufgaben bearbeiten aber jeweils eine einzelne Arbeitsanforderung von außen.

Man muß sich den Ablauf im groben folgendermaßen vorstellen:

1. Der Benutzer führt eine Aktion durch. Dies kann zum Beispiel ein Tastaturdruck sein.
2. Dieser Tastaturdruck führt zu einer Menge von Anforderungen an den Rechner. So muß die gedrückte Taste erkannt, die Darstellung der entsprechenden Taste auf dem Bildschirm ausgegeben und eventuell durch diesen Tastaturdruck verursachte Aufgaben im Anwendungsprogramm ausgeführt werden.
3. Alle diese Aufgaben führen zu einer Flut von Operationen, von denen eine schnelle CPU mehrere Millionen pro Sekunde durchführen kann.
4. Alle diese Operationen werden nun von der CPU entgegengenommen und sequentiell, das heißt schön der Reihe nach, abgearbeitet.

Eine moderne CPU kann zwar mehrere Operationen gleichzeitig lesen, aber in den seltensten Fällen gleichzeitig durchführen. Das gleichzeitige Lesen mehrerer Operationen dient lediglich der Entlastung des Rechners, da dieser das Lesen eines Schwungs von Operationen immer schneller durchführen kann als das einzelne Lesen aller dieser Operationen. Das wird durch die folgende Kommunikation versinnbildlicht:

– Bei der Beschaffung mehrerer Operationen wird von der CPU folgende Anforderung an den Rechner gestellt: »Gib mir bitte 100 Operationen«. Der Rechner liefert diese Operationen mit den Worten »Hier hast du deine 100 Operationen«.

– Die Kommunikation bei der Beschaffung einzelner Operationen würde in etwa wie folgt aussehen »Gib mir bitte eine Operation«, »Hier hast du eine Operation«, »Gib mir bitte eine Operation«, »Hier hast du eine Operation«, und so weiter.

Es dürfte einleuchtend sein, daß der erste Schritt dem zweiten jederzeit vorzuziehen ist.

Wie kann denn nun ein Computer mit einer CPU, die in der Regel immer nur eine Aufgabe bewältigen kann, und einem einzelnen Benutzer, mehrere Dinge gleichzeitig machen?

Ziehen wir zur Beantwortung dieser Frage einmal eine gängige Textverarbeitung und eine Tabellenkalkulation heran.

Der Benutzer hat in seiner Tabellenkalkulation eine umfangreiche Kalkulation durchzuführen und möchte nicht tatenlos auf das Ergebnis dieser Berechnung warten. Was liegt da näher, als die Wartezeit mit etwas Sinnvollem zu verbringen, wie zum Beispiel einen Brief zu schreiben. Das würde bedeuten, daß beide Programme, Tabellenkalkulation sowie Textverarbeitung,

»gleichzeitig« aktiv sind. Während das eine Programm, die Tabellenkalkulation, mit sich selbst beschäftigt ist, schreibt der Benutzer in der anderen Anwendung, der Textverarbeitung, einen Brief. Nun haben wir aber eingangs festgestellt, daß ein Computer in der Regel eben doch nur eine Aufgabe gleichzeitig bewältigen kann.

Aus diesem Dilemma befreit sich der Computer mit einem Trick. Jede der beiden Anwendungen erhält immer ein wenig Rechenzeit zugewiesen, man spricht hier von Zeitscheiben. Die Zuweisung der Rechenzeit erfolgt so schnell und in ganz kleinen Happen, so daß der Benutzer nicht merkt, daß eine der beiden Anwendungen gerade schläft, während die andere arbeitet.

Würden wir der Textverarbeitung aus dem Beispiel die gleiche Rechenzeit zugestehen wie der Tabellenkalkulation, also 50%, so würde dies der Benutzer wahrscheinlich merken, denn eine um 50% abgesenkte Aufmerksamkeit des Systems kann nicht vor dem Benutzer geheimgehalten werden. Eine der wichtigen Aufgaben des Betriebssystems ist es nun, die Verteilung der Rechenzeit unter den aktiven Anwendungen derart geschickt zu verteilen, daß dem Benutzer die praktisch parallele Verarbeitung aller Aufgaben vorgegaukelt wird.

In dieses Aufgabengebiet mit einbezogen werden müssen die folgenden Faktoren:

- Eine Vordergrundanwendung, das ist das Programm, in dem der Benutzer gerade Eingaben vornimmt, muß eine etwas erhöhte Aufmerksamkeit erhalten.
- Hintergrundprogramme, das sind die Programme, die gerade mit eigenen Aufgaben beschäftigt sind, müssen nicht mit großer Aufmerksamkeit bedacht werden.
- Programme, die gerade mit Datentransfer beschäftigt sind, das kann eine Disketten-, Festplatten- oder gar Modemkommunikation sein, benötigen wie die Vordergrundanwendung eine etwas gesteigerte Aufmerksamkeit. Dies wird vor allen Dingen praktiziert, um die Rechner schnellstens aus dem Verkehrsstau beim Transfer zu befreien.
- Beim benutzergesteuerten Wechsel von Anwendungen, das heißt, der Benutzer wendet sich zum Beispiel von der Textverarbeitung doch wieder der Tabellenkalkulation zu, erhält die Anwendung, die in den Vordergrund gelangt, einen kleinen Extraschub.

Alle diese Dinge werden von einem Teil des Betriebssystems verwaltet, der als Scheduler (auch Dispatcher) bezeichnet wird. Dieser hat jederzeit die absolute Kontrolle über das System, und seine Aufgabe besteht im Wesentlichen in der automatischen Zuteilung von Rechenzeit an die aktiven Anwendungen. Der Scheduler ist Bestandteil aller Multitasking-fähigen Betriebssysteme.

Alle aktiven Anwendungen werden vom Scheduler in eine sogenannte Warteschlange eingestellt, die der Scheduler beobachtet und verwaltet. Als Informationen merkt sich der Scheduler den gesamten Zustand aller Register der CPU zum Zeitpunkt der Unterbrechung einer zuvor aktiven Applikation (Applikationskontext). In diesen CPU-Registern wiederum enthalten ist unter anderem der Zeiger auf die nächste auszuführende Programmstelle. Wird einer zuvor unterbrochenen Anwendung wieder eine Zeitscheibe zugeordnet, so werden alle Register in der CPU wieder in den Zustand versetzt, der bei der Unterbrechung galt. Die Anwendung wird an der Stelle fortgeführt, an der sie zum Zeitpunkt der Unterbrechung aktiv war. Weder die Anwendung noch der Benutzer merken auf diese Weise etwas von der Unterbrechung.

Ältere Systeme haben die Wichtigkeit der Anwendungen in einem Multitaskingsystem mit einer Wertigkeitentabelle verteilt. Üblicherweise zu vergeben waren 1 bis 255 Wertigkeitspunkte, auch als Prioritäten bezeichnet. Jede Anwendung erhielt eine bestimmte Wertigkeit, wobei 1 die höchste Wertigkeit und 255 die niedrigste Wertigkeit war. Eine Anwendung mit der Wertigkeit 0 war gerade aktiv, alle anderen Anwendungen warteten auf ihre Ausführung. Wurde die Zeitscheibe der aktiven Anwendung beendet, üblicherweise nach einer bestimmten Anzahl Systemtakte, dann wurde die Wertigkeit der gerade unterbrochenen Anwendung wieder auf ihren Startwert gesetzt. Bei einem Suchlauf durch die wartenden Anwendungen wurde die aktuelle Wertigkeit aller Anwendungen um 1 vermindert. Die Anwendung, deren Priorität dabei auf 0 gesetzt wurde, erhielt dann die nächste Zeitscheibe und wurde aktiviert. Hieraus wird auch ersichtlich, warum die Anwendung mit der Priorität 255 die am wenigsten berücksichtigte Anwendung war. Sie benötigte immer 255 Scheduler-Suchläufe, bis ihre Wertigkeit auf 0 herabgesunken war und somit zur Aktivierung anstand. Die Anwendung mit der Wertigkeit 1 hatte es dagegen besser, sie wurde in fast jedem zweiten Suchlauf zur Aktivierung gebracht.

Dieses alte Konzept hatte einige Schwachpunkte. Zum einen wurde zu keinem Zeitpunkt berücksichtigt, ob eine Anwendung im Vorder- oder Hintergrund war. Die Folge waren ruckelnde und nicht aufmerksame Vordergrundanwendungen. Zum anderen muß immer in einem Multitaskingsystem den Anwendungen die Möglichkeit eingeräumt werden, ihre eigene Wertigkeit anpassen zu können. Die Folge war, daß alle Anwendungen zum Start ihre Priorität auf 1 setzten, um mit einer optimalen Antwortzeit glänzen zu können. Da dies fast alle Anwendungen machten, wurde dieses Konzept ad absurdum geführt und im Laufe der folgenden Betriebssystemgenerationen verbessert.

OS/2 Warp 4 geht hier einen anderen Weg. An dem Konzept der Zeitscheiben und dem Scheduler als Kontrollinstanz hat sich nichts geändert, verbessert hingegen wurde die Einstellbarkeit der Prioritäten sowie die Wirksamkeit des Schedulers auf Thread-Ebene anstelle auf Anwendungsebene.

Moderne Betriebssysteme unterscheiden zwischen zwei grundlegend verschiedenen Verteilungsmechanismen bei der Zuordnung von Rechenzeit:

– Das präemptive Multitasking-Konzept hat jederzeit die Kontrolle über das System und weist den aktiven Anwendungen die Rechenzeit zu, die der Scheduler für angemessen hält. Die einzelnen Anwendungen haben nur wenig Einfluß auf die Einordnung in dieses Schema.

– Das kooperative Multitasking-Konzept verläßt sich auf die Kooperation der Anwendungen untereinander. Jede Anwendung teilt dem System mit, ab wann andere Anwendungen wieder aktiv werden können. Damit nicht eine einzelne fehlgesteuerte Anwendung die gesamte Rechenzeit beansprucht, existiert eine Kontrollinstanz, die nur in Ausnahmesituationen regelnd eingreift.

OS/2 Warp 4 ist ein präemptives Mutitasking-System. Der Scheduler hat immer die absolute Kontrolle über das System und kann jederzeit die Zeitscheibe einer aktiven Anwendung unterbrechen und an eine andere Anwendung weiterreichen. Der Scheduler macht hierbei keine Ausnahme und unterbricht, nach Ablauf der jeweilige Zeitscheibe, sogar atomare Befehle. Als atomare Befehle werden die Programmierzeilen von Hochsprachen wie C oder C++ genannt.

Jede Programmierzeile in einer Hochsprache wird von einem Übersetzungsprogramm in eine unterschiedliche Anzahl von Assembler-Operationen für die CPU übersetzt. Der Scheduler des OS/2 Warp 4 unterbricht auf Operationen-Ebene, das bedeutet, daß eine Anwendung mitten in der Ausführung eines C-Befehles unterbrochen werden kann.

Dies hört sich schlimmer an als es ist, denn es müssen von Anwendungsseite keine Vorbeugemaßnahmen ergriffen werden, der Scheduler merkt sich den exakten Zustand einer unterbrochenen Anwendung und setzt auf diesem exakten Zustand bei der Erteilung der nächsten Zeitscheibe wieder auf.

Betriebssysteme der zweiten Kategorie, zum Beispiel Windows, sind sehr von der fairen Kooperation aller Anwendungen abhängig. Je mehr schlecht programmierte Anwendungen aktiv sind, desto öfter muß die Kontrollinstanz einschreiten. Da dieses Einschreiten wiederum Rechenzeit kostet, kann der Durchsatz solcher Betriebssysteme sehr schnell sinken. Ist nur eine Anwendung aktiv, ist dagegen die Rechenzeit meist besser als bei den präemptiven Multitasking-Betriebssystemen. Deshalb wird oft empfohlen, bei Systemen mit hoher Systemlast auf präemptive Betriebssysteme zurückzugreifen und die kooperativen Betriebssysteme dann einzusetzen, wenn die Benutzer nur mit einer Aufgabe befaßt sind.

2.2.2 Multithreading

Während beim Multitasking die Fähigkeit beschrieben wird, verschiedene Anwendungen (Prozesse) »gleichzeitig« ausführen zu können, so wird mit Multithreading die Fähigkeit bezeichnet, innerhalb einer einzelnen Anwendung mehrere Aufgaben (Threads) parallel ausführen zu können.

Ziehen wir als Beispiel wieder die Textverarbeitung heran, so ergibt sich hier oft die Anforderung, daß manche Benutzer einen Text ausdrucken wollen, um danach mit der Eingabe eines anderen Texts fortzufahren. Diese Aufgabe kann aus einem Programm nur dann gleichzeitig gelöst werden, wenn das Programm die Druckaufgabe parallel löst, während dem Benutzer das Eingabefenster zur Erfassung eines weiteren Texts weiterhin zur Verfügung steht.

Die parallele Abarbeitung von Aufgaben innerhalb einer Anwendung läßt sich nur mit Hilfe von Threads bewerkstelligen.

OS/2 Warp 4 unterscheidet vier Typen von Threadzuständen. Diese, in abgestufter Wertigkeit, werden benannt mit:

– Time critical (zeitkritisch)

– Server (Verteilungsanwendung)

– Regular (normale Anwendung)

– Idle (nur, wenn nichts Wichtigeres zu tun ist)

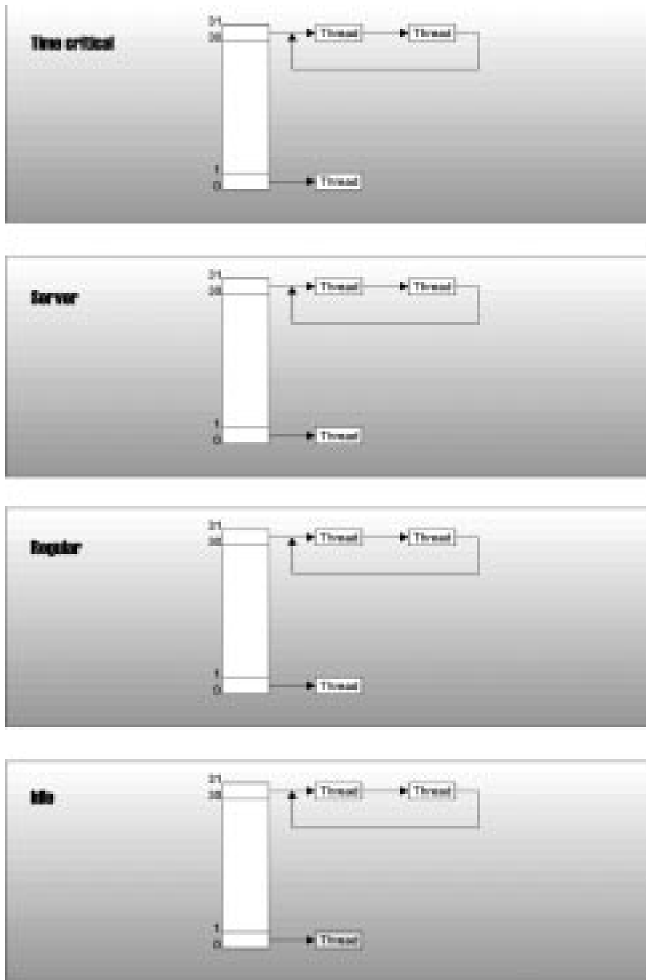


Abb. 2.3: 4 Prioritätsklassen mit jeweils 32 Prioritätsstufen, ausgerüstet mit jeweils einer Warteschlange

In jede diese vier Prioritätsstufen werden die aktiven Threads eingestuft. Jede dieser Prioritätsstufen kennt wiederum Wertigkeiten von 0 bis 31. Jede dieser Wertigkeiten wiederum hat eine eigene Warteschlange, die als sogenannte Round Robin funktioniert. Round Robin steht für eine Endlosschleife und bedeutet, daß der Thread eine Zeitscheibe erhält, der in der Warteschlange an erster Stelle angelangt ist. Ein unterbrochener Thread wird wieder ans Ende der Schlange eingeordnet. Hieraus wird ersichtlich, daß der Scheduler des OS/2 Warp 4 mit insgesamt 128 Warteschlangen ausgerüstet ist (vier Prioritätsstufen mit jeweils 32 Wertigkeiten, ausgerüstet mit je einer Warteschlange). Dieses System wurde bereits für zukünftige Generationen geplant.

OS/2 Warp 4 ist ein Multitasking- und Multithreading-Betriebssystem.

2.3 Interprozeß-Kommunikation

Zwei Probleme stellen sich in einer Multitasking- und Multithreading-Umgebung. Zum einen muß in einem strikt abgeriegelten Speichermodell wie dem von OS/2 der Zugriff auf gemeinsam genutzte Ressourcen über Prozeßgrenzen hinweg ermöglicht werden. Zum anderen bedarf der Zugriff auf gemeinsame Ressourcen innerhalb eines Prozesses durch mehrere Threads der Synchronisation dieser Zugriffe.

Grundsätzlich geht es in diesem Kapitel um den Informationsaustausch zwischen unterschiedlichen Programmen und zwischen unterschiedlichen Programmteilen, die auf gemeinsam genutzte Systemressourcen zugreifen müssen.

IBM OS/2 Warp 4 kennt und unterstützt alle gängigen Arten der Interprozeß-Kommunikation. Dabei bietet sich jedes Kommunikationsmodell für eine bestimmte Aufgabenstellung an. Wichtig für den Entwickler ist, daß man die Wirkung aller Modelle kennt, um dann im Bedarfsfall auch das richtige Modell zu wählen. Mitunter kann die eine oder andere Spielart temporär das gleiche Ergebnis erzielen, aber für jede Anforderung ist immer nur eine Lösung die beste.

Entgegen landläufiger Praxis nehmen wir in diesem Kapitel die Sonderformen der Interprozeß-Kommunikation mit auf. Der Presentation Manager, ein Bestandteil des OS/2, bietet »seinen« Applikationen einige besondere Möglichkeiten der Kommunikation.

2.3.1 Clipboard

Dieses Protokoll wird von dem Presentation Manager bereitgestellt.

Das Clipboard ist ein Speicherbereich, der vom Presentation Manager für alle Anwendungen zur Verfügung gestellt wird. Dieser Speicherbereich existiert nur einmal und ist als Zwischenspeicher für die benutzergesteuerte Übertragung von Daten vorgesehen.

Benutzergesteuert bedeutet in diesem Fall, daß den Benutzern über die Menüpunkte *Kopieren, Ausschneiden* und *Einfügen* das Manipulieren des Clipboards ermöglicht wird.

Der Benutzer selektiert in einer Anwendung, wie zum Beispiel einem Editor, einen Textausschnitt und überträgt mit dem Menüpunkt *Kopieren* diesen selektierten Text in das Clipboard. Der selektierte Text im Editor bleibt bei dieser Aktion unverändert. Bei Nutzung des Menüpunkts *Ausschneiden* wäre der Text im Editor entfernt worden.

Nach Übertragung des Textes kann der Benutzer eine andere Anwendung in den Vordergrund holen, es kann sich dabei auch um die selbe Anwendung handeln, und dort den Text an einer zuvor markierten Stelle wieder einfügen. Hierzu betätigt er den Menüpunkt *Einfügen,* und der im Clipboard enthaltene Text wird an der markierten Stelle erscheinen.

Mit einigen APIs wie *WinSetClipbrdData* oder *WinQueryClipbrdData* können PM-Anwendungsprogramme Daten in das Clipboard übertragen oder auch dort wieder abfragen. Anwendungen sollten auf keinen Fall die Daten im Clipboard automatisch manipulieren, sondern dies ausschließlich auf Benutzeraktion vornehmen.

Wie Sie sicher schon bemerkt haben, wird der Menüpunkt *Einfügen* in jeder Anwendung erst zu dem Zeitpunkt zur Verfügung gestellt, sobald in einer anderen beliebigen Anwendung die Aktion *Kopieren* oder *Ausschneiden* angewählt wurde. Dies wird ermöglicht durch ein sogenanntes »Clipboard Viewer Event«, das allen aktiven Presentation-Manager-Anwendungen auf Wunsch übermittelt wird, sobald der Clipboard-Inhalt verändert wird.

Das Clipboard ist zur Zeit immer noch auf 64 Kbyte Daten beschränkt und kann diverse Datenformate wie Text, Metafile, Bitmap und einige andere mehr enthalten. Das Clipboard kann immer nur ein Element eines Datenformats zu jeder Zeit enthalten.

Der Datenaustausch von Clipboard-Inhalten zwischen OS/2 und Win-OS2 Anwendungen ist möglich.

2.3.2 Direct Manipulation Protocol (Drag and Drop)

Dieses Protokoll wird ebenfalls vom Presentation Manager bereitgestellt.

Sobald auf der OS/2-Oberfläche Objekte vom Benutzer verschoben werden, ist das Drag-and-Drop-(DnD)Protokoll involviert. Die Kommunikation zwischen den beteiligten Anwendungen erfolgt mit Hilfe von PM-Events. Das sind Nachrichten, die über die Message-Queue des Presentation Managers an betroffene Anwendungen verteilt werden.

DnD kennt im eigentlichen Sinne keinen Datenlieferanten (Server) oder einen Datenempfänger (Client), sondern nur an der Kommunikation beteiligte Fenster. Drag and Drop wird aktiv, sobald ein Objekt mit der rechten Maustaste angeklickt wird und, während die Maustaste gedrückt bleibt, das Objekt verschoben wird. In diesem Moment wird das Fenster, auf dem diese Aktion gestartet wurde, das Herkunftsfenster. Während der Überquerung anderer Fenster mit der gedrückten Maustaste wird jedes überquerte Fenster mit Events darauf hingewiesen, daß es soeben von einem Objekt überquert wird. Die überquerten Fenster haben nun die Möglichkeit, die in den Drag-and-Drop-Events gekapselten Daten zu überprüfen und zu entscheiden, ob ein Drop, das ist das Fallenlassen des Objekts, sinnvoll wäre.

Verneint das überquerte Fenster den Drop, so erscheint auf der Oberfläche ein Sperrsysmbol. Dabei kann die Ablehnung grundsätzlich oder auch nur temporär erfolgen. Bei grundsätzlicher Ablehnung wird das Fenster während dieser Drag-and-Drop-Operation nicht mehr informiert. Bei der temporären Ablehnung teilt man dem System hingegen mit, daß man grundsätzliches Interesse hat, aber im Moment die Datenlieferung ablehnen muß.

Läßt der Benutzer die Maustaste auf einem Fenster los, auf dem der Drop erlaubt ist, so gilt dieses Fenster als Zielfenster.

Nun liegt es in der Natur des PM, daß vom Drag-and-Drop-Vorgang mehrere Anwendungen oder auch nur eine einzelne Anwendung betroffen sein können. Aus diesem Grund müssen die Daten vor der Überprüfung oder auch vor der Entgegennahme in Zugriff gebracht werden, da sonst der Speicherschutz zuschlägt. Das DnD-Protokoll transportiert die Daten in speziellen Speicherbereiche, die von jeder betroffenen Anwendung vor dem Zugriff entriegelt werden müssen. Für die Entriegelung der Speicherbereiche stellt das Drag-and-Drop-Protokoll spezielle API-Befehle zur Verfügung.

2.3.3 Dynamic Data Exchange (DDE)

Dieses Protokoll wird vom Presentation Manager bereitgestellt.

Dynamic Data Exchange erlaubt die Kommunikation zwischen Anwendungen mit der Hilfe von PM-Events. Das DDE-Protokoll bietet DDE-fähigen Applikationen die Abfrage des Datenstatus anderer Applikationen. Dabei kann der Datenaustausch einmalig mit Hilfe einer dynamisch zu erzeugenden Verbindung geschehen, als auch mittels eines permanenten Links zwischen zwei Applikationen.

Als gemischtes Protokoll kann der Verbindungsaufbau sowohl durch Benutzerinteraktion als auch vollkommen automatisch zwischen zwei DDE-fähigen Applikationen ohne Benutzerinteraktion erfolgen.

DDE unterscheidet zwischen einem DDE-Server, das ist die Anwendung, die ihre Daten anderen Anwendungen zur Verfügung stellen kann, und einem DDE-Client, der von einem DDE-Server Daten erhalten möchte. Ein DDE-Server kann beliebig viele DDE-Clients gleichzeitig mit unterschiedlichen Datenlieferungen befriedigen. Es können mehrere DDE-Server gleichzeitig aktiv sein. Weiterhin kann ein Client gleichzeitig mit mehreren DDE-Servern kommunizieren.

Ein Verbindungsaufbau geschieht immer vom Client an den Server.

Mit Hilfe ausgeklügelter, schlüsselwortbasierter Protokolle haben die am DDE teilnehmenden Anwendungen jederzeit die Möglichkeit, die aktuellen Daten des DDE-Servers zu erfragen. Darüberhinaus bietet das DDE-Protokoll die Abfrage aller im System aktiven DDE-Server und deren Fähigkeiten.

Bei der einfachen, dynamisch aufzubauenden Verbindung fragt der Client zunächst den Server, ob dieser ein bestimmtes Datum zur Verfügung stellen kann. Wird dies vom DDE-Server positiv bestätigt, so wird eine temporäre Verbindung aufgebaut, in deren Ablauf eine Information zwischen dem Server und dem Client ausgetauscht werden kann. Dabei sind beide Richtungen der Datenlieferung erlaubt. Im Falle einer Datenlieferung vom Server an den Client spricht das DDE-Protokoll von einem *REQUEST*. Schickt ein Client hingegen Daten an den Server, so spricht man in diesem Fall von einem *POST*.

DDE kennt aber auch permanente, feste DDE-Verbindungen zwischen Anwendungen oder auch einzelnen Anwendungsbestandteilen. Eine permanente DDE-Link wird nach Anfrage vom Client an den Server aufgebaut. Dabei wird hier wiederum zwischen zwei Handlungsarten unterschieden. Ändern sich die Daten beim Server, so kann entweder das geänderte Datum automatisch an den Client geschickt werden oder nur ein Hinweis, daß neue Daten vorliegen. Im letzteren Fall muß sich der Client die geänderten Daten explizit abholen, dies geschieht dann immer noch über die existierende permanente Verbindung.

Da die im DDE-Protokoll enthaltenen Daten Applikationsgrenzen überschreiten können, muß jede Anwendung vor dem Zugriff auf diese Daten den Speicherschutz entriegeln. Dies geschieht allerdings nur mit dafür vorgesehenen speziellen API-Befehlen, so daß die Anwendungen das Speicherschutzprinzip des OS/2 nicht unterwandern können.

OS/2 und Win-OS2 Anwendungen können über das DDE-Protokoll miteinander kommunizieren.

2.3.4 Exceptions

Exceptions sind sogenannte Ausnahmebedingungen, die im Ablauf einer Anwendung immer schon mal auftauchen können. So gibt es Exceptions für den Fall, daß auf Speicherbereiche zugegriffen wurde, die nicht im Kontext der Applikation enthalten sind, oder auch Exceptions für den Zugriff auf ein CD-Laufwerk, in dem keine CD eingelegt wurde.

Für die Bearbeitung einer Exception ist zunächst ausschließlich das Betriebssystem zuständig. Die Standardverarbeitung des Systems unterscheidet sich von Exception zu Exception, läuft aber meist auf den »Abschuß« der Applikation hinaus.

OS/2-Applikationen haben die Möglichkeit, eigene Exception-Handler zu registrieren, die vor der Standardverarbeitung zum Tragen kommen. So kann zum Beispiel die Meldung »Keine CD im Laufwerk enthalten« unterbunden werden. Bei schwereren Verfehlungen hat ein registrierter Exception-Handler nur die Möglichkeit, Aufräumarbeiten durchzuführen, aber am Abschuß der Anwendung läßt sich nicht rütteln.

Exception-Handler können auf Thread-Basis registriert werden. Registrierte Handler werden vom Betriebssystem in einer Kette eingereiht, die dann im Falle eines Falles schön der Reihe nach abgearbeitet wird.

Der schlimmste aller Fehler ist die »General Protection Fault Exception« (GPF). Eine GPF wird immer dann erzeugt, wenn ein Anwendungsprogramm auf einen Speicherbereich zugreifen will, der außerhalb seines zugewiesenen Kontexts liegt. Selbst diese Exception kann mit einer registrierten Funktion hinterlegt werden, am Abschuß der Anwendung kann diese aber nichts mehr ändern, es sei denn, diese Funktion besorgt sich wiederum freien Speicher und biegt den Zugriff von der falschen Seite auf eine gültige Seite um. Dieses Verfahren gehört aber zu der Sorte »schmutzige Tricks« und wird eigentlich nicht angewendet.

Eine spezielle Exception ist der sogenannte »Guard Page Fault«. Wie bereits aufgeführt, werden die angeforderten Stackbereiche von Secondary Threads nicht direkt vom System bereitgestellt, sondern nur in dem Fall, wenn diese tatsächlich benutzt werden. OS/2 markiert die erste Seite eines solchen Bereichs als sogenannte Guard-Page. Wird diese Aufpasserseite vom Thread angefaßt, so erfolgt der Guard-Page-Hinweis an das System. Das System wiederum besorgt sich schnell eine leere Seite und blendet diese an der angeforderten Stelle ein, so daß der Thread darauf zugreifen kann. Anschließend wird die Seite hinter dieser Seite als Guard-Page markiert. Eine Guard-Page ist somit eine Seite, die zwar da ist, aber doch nicht ganz da ist.

Eine weitere Besonderheit der Guard-Pages betrifft die Anwendungsentwickler. Greift ein Anwendungsprogramm nicht auf die bereitgestellte Guard-Page zu, sondern direkt auf eine Speicherstelle, die in einer anderen Seite hinter der Guard-Page liegt, dann wird wertvoller Speicher vergeudet. Die original Guard-Page bleibt eine Guard-Page und wartet weiter auf einen Zugriff. Der Zugriff auf die Speicherstelle wird analysiert und als gültig erkannt. Diese Seite wird besorgt und dem Anwendungsprogramm zur Verfügung gestellt. Die Seite dahinter wird zur neuen Guard-Page – und schon existieren zwei Guard-Pages im selben Stack. Moderne Compiler berücksichtigen diesen Umstand und korrigieren solche Zugriffe durch sogenannte »Funktionsprologe«, das sind vorgeschaltete Korrekturfunktionen. Diese sorgen dann dafür,

daß alle noch nicht freigegebenen Seiten bis zu der angesprochenen Speicherstelle physikalisch realisiert. Das System spricht hier von Stack-Probes, die einen sauberen Umgang mit Guard-Pages auf dem Stack garantieren.

2.3.5 Pipes, Named Pipes

Die deutsche Übersetzung von »Pipe« eignet sich hervorragend zur Erklärung. Eine Pipe ist ein Rohr, in dem man an einem Ende etwas reinstecken kann, was dann am anderen Ende wieder auftaucht.

Der Pufferbereich der Pipes ist nicht unendlich groß, sondern das Beschreiben des Puffers beginnt wieder von vorne, wenn der Puffer voll ist. Daraus folgt, daß die Daten möglichst schnell wieder entfernt werden, da sonst die Pipe nicht mehr nutzbar wird. Die Größe des Puffers kann festgelegt werden.

OS/2 unterscheidet zwischen zwei Arten von Pipes. Es gibt »unbenannte Pipes« (anonymous) und »benannte Pipes« (Named). Die unbenannten Pipes eignen sich für die Kommunikation von Prozessen, die einander bekannt sind. Üblicherweise handelt es sich hierbei auch um Prozesse, die auf einer Maschine, man spricht hier von lokal, ablaufen. Die Puffergröße der unbenannten Pipes ist limitiert, und der Zugriff auf die Daten kann immer nur in eine Richtung erfolgen.

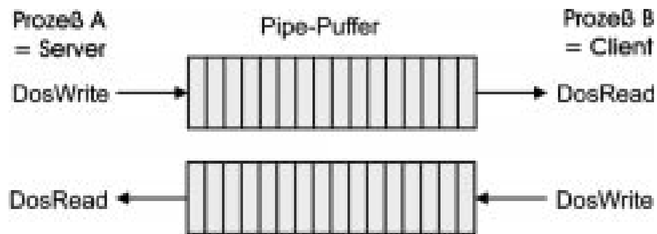


Abb. 2.4: Datenaustausch mit Hilfe von Pipes

Die benannten Pipes eignen sich hervorragend für die Kommunikation von nicht näher verwandten Prozessen. Diese Prozesse können sowohl auf einer lokalen Maschine ablaufen, als auch auf voneinander entfernten Maschinen. Weiterhin können sich die miteinander kommunizierenden Prozesse, die Named Pipes verwenden, auch auf unterschiedlichen Plattformen befinden, da Named Pipes von einer ganzen Reihe Betriebssysteme unterstützt werden. Named Pipes sind weitaus flexibler in der Definition der Größe des Puffers und erlauben das Lesen und Schreiben in beide Richtungen. Beim Anlegen einer solchen Pipe wird in Wahrheit nicht nur ein zirkulierender Puffer angelegt, sondern gleich zwei. Jede Schreib-Lese-Richtung erhält einen eigenen Puffer. Für das Anlegen einer solchen komplexen Pipe gibt es beim Öffnen von Named Pipes eine Vielzahl von beschreibenden Optionen. Die Anlage einer vollduplex-fähigen Named Pipe ist eine davon.

Auch bei den Pipes unterscheidet OS/2 zwischen sogenannten Server-Prozessen und Client-Prozessen. Der Server ist die Anwendung, die die Pipe eingerichtet hat. Hierfür gibt es die API-Befehle *DosCreatePipe* (Anonymous Pipe) sowie *DosCreateNPipe* (Named Pipe).

Mit den normalen Schreib- und Lesebefehlen des OS/2-APIs kann eine angelegte Pipe mit Daten gefüttert werden und die Daten auch wieder aus ihr ausgelesen werden. Dies trifft insbesondere für die Anonymous Pipes zu. Der Benutzer kann das Wirken von Pipes sehr schön auf der Kommandozeile betrachten, wenn er den Pipe-Befehl »|« anwendet. So wird bei der folgenden Kommandozeile eine Pipe angewendet:

DIR *.* | MORE

Der DIR-Befehl verschickt die Daten, und MORE nimmt diese Daten zwecks Aufbereitung entgegen. Der DIR schickt als Daten die Verzeichnisdaten über den Standardausgabestream. Dieser Ausgabestream wird statt auf dem Bildschrim in die Pipe umgeleitet. MORE entnimmt diese Daten und zeigt diese nur seitenweise an. Der Benutzer muß dann pro angezeigte Seite einmal die Enter-Taste drücken. Da der DIR-Befehl die Daten mit Sicherheit schneller liefern kann, als der Benutzer diese Daten auf dem Bildschirm mit [Enter] seitenweise bestätigen kann, wird hier schnell ersichtlich, daß eine Pipe die betroffenen Prozesse in einen Wartezustand versetzen kann. Es ist nicht anzunehmen, daß der Pipe-Puffer so groß gewählt wurde, daß immer alle Daten des DIR-Befehles in die Pipe passen können.

Werden die Daten auf der einen Seite der Pipe nicht so schnell entnommen, wie sie auf der anderen Seite in die Pipe eingestellt wurden, so wird der Sender der Daten in einen »Waiting for I/O«-Status versetzt. Er kann erst dann wieder in die Pipe schreiben, wenn wieder Platz in der Pipe vorhanden ist.

Auch ein Client-Prozeß kann in einen »Waiting for I/O«-Status versetzt werden, sobald der Server nicht mit der Datenlieferung nachkommen kann.

Den »Waiting for I/O«-Status haben wir bereits im Kapitel »Multitasking und Multithreading« besprochen. Einem Thread eines Prozesses, der in einen »Waiting for I/O«-Status versetzt wurde, werden keine Zeitscheiben zugewiesen, er beansprucht deshalb keine Systemzeit. Wird die Pipe für ihn wieder freigegeben, so wird er mit ziemlicher Sicherheit sofort aktiviert werden und er erhält als Ausgleich für die lange Wartezeit einen entsprechenden I/O-Burst.

Während das Protokoll für Anonymous Pipes sehr einfach zu handhaben ist, zeigt sich der Umfang der Features bei den Named Pipes ungleich umfangreicher. Named Pipes erlauben das Zusammenfassen von mehreren Pipes über Prozesse und Rechner hinweg. Hierzu existieren dann separate API-Aufrufe, die ein komplettes Transaktions-Management mit Pipes aufbauen lassen. So kann eine Transaktion zum Beispiel aus dem erfolgreichen Abarbeiten von mehreren Daten aus mehreren Pipes bestehen. Erst wenn alle diese Daten erfolgreich bearbeitet wurden, werden die Daten aus der Pipe entfernt. Weiterhin gibt es auch den Lesezugriff auf Daten. In der Pipe-Terminologie bedeutet dies »Kopieren«, die Daten werden entnommen, aber nicht aus der Pipe entfernt.

Named Pipes müssen mit einem Namen versehen werden. Dieser Name muß mit einem Präfix beginnen, der »\PIPE\« lautet (Beispiel \PIPE\MeinePipe). Eine solche Pipe befindet sich auf dem lokalen Rechner. Bei dem Zugriff auf Pipes, die von einem entfernten Rechner herangezogen werden sollen, muß ein weiteres Präfix vorgesetzt werden, das den Computernamen

enthält (Beispiel \\Server\PIPE\MeinePipe). Beachten Sie bitte die beiden Schrägstriche zu Beginn. Diese Art der Namensgebung ist den UNC-Namen angelehnt, die bei Netzwerken zur Anwendung kommen. Nach einem Doppelslash folgt immer ein Computername. Das Schlüsselwort »PIPE« ist ein Devicename, für den eine eigene Tabelle in jedem System existiert.

OS/2 unterscheidet vier Pipe-Zustände:

- *Connected*. Die Pipe wurde von einem Server-Prozeß angelegt und ein Client hat sich bereits als Nutzer angemeldet. Das Lesen und Schreiben in die Pipe ist freigegeben.
- *Closing*. Die Pipe befindet sich in einem unbestimmten Zustand. Ein Client hat sich abgemeldet, aber der Server hat dies noch nicht bestätigt.
- *Disconnected*. Ein Server-Prozeß hat die Pipe angelegt, aber sich noch nicht angebunden. Als andere Ursache kann dieser Zustand dadurch entshehen, daß der Server die Pipe abgemeldet hat. Das Öffnen der Pipe durch andere Prozesse ist nicht möglich.
- *Listening*. Die Pipe wurde vom Server angelegt und angebunden. Allerdings hat sich noch kein Client für die Benutzung angemeldet.

Pipes, und hier vor allen Dingen die Named Pipes, sind das mächtigste und wichtigste Interprozeß-Kommunikationsprotokoll, das OS/2 zu bieten hat. Durch seine Flexibilität werden alle denkbaren Aufgaben der Interprozeß-Kommunikation ermöglicht. Vor dem Einsatz von Pipes sollte allerdings zunächst überprüft werden, ob nicht ein anderer, einfacherer Kommunikationsweg eingeschlagen werden kann, da Pipes für kleinere Kommunikationsaufgaben etwas zu mächtig sind.

2.3.6 Queues

Queues sind eine etwas abgeschwächte Form von Pipes.

Eine Queue, auch als Kette bezeichnet, ist immer benannt und kennt immer nur einen Server, dafür aber eine beliebige Anzahl Clients. Der Datenstrom in einer Queue fließt immer nur in eine Richtung, nämlich von den Clients zu dem Server. Der Name einer Queue startet immer mit dem Präfix »\QUEUES\« (Beispiel \QUEUES\MeineQueue). Das »QUEUES«-Präfix deutet auch hier wie bei den Pipes auf eine Device-Tabelle, die in jedem OS/2-Rechner vorgehalten wird. Queues können nur lokal verwendet werden.

Das Anlegen einer Queue erfolgt mit einem speziell hierfür vorgesehenen API-Befehl mit dem Namen *DosCreateQueue*. Das Schreiben und Lesen von Queue-Daten erfolgt mit den *DosWriteQueue*- und *DosReadQueue*-APIs.

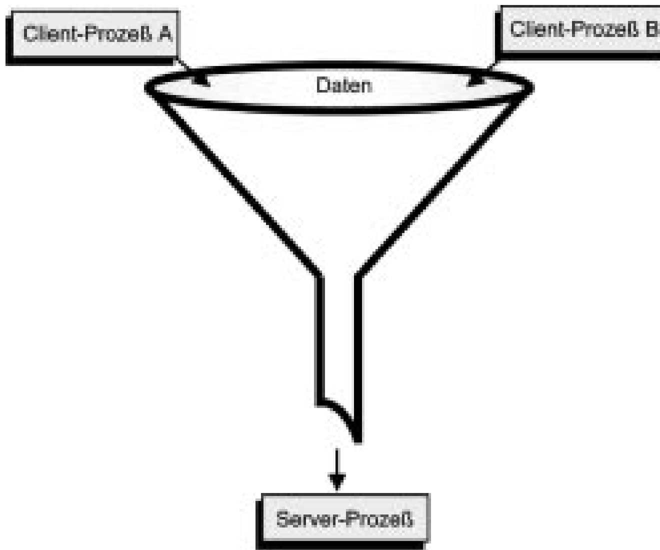


Abb. 2.5: Queues sind Datensammler für einen Server-Prozeß

Das Queue-Protokoll erlaubt das Feststellen der Anzahl aller Elemente in der Queue, sowie das schnelle Löschen aller Elemente.

Typische OS/2-Anwendungen nutzen Queues in zwei unterschiedlichen Formen. Entweder werden die Daten selbst in der Queue transportiert, oder es werden nur Zeiger oder Hinweise auf die Daten durch die Queue geschleust. Das Einstellen der Daten selbst ist nur für die Anwendungen praktikabel, die mit geringen Datenmengen arbeiten. Erreicht das Datenvolumen aber etwas größere Ausmaße, so sollte statt der eigentlichen Daten nur ein Zeiger auf die Daten durch die Queue transportiert werden. Dies spart die zeitintensiven Kopierfunktionen in und aus der Queue. Allerdings muß in diesem speziellen Fall berücksichtigt werden, daß der Zugriff auf die Daten, deren Pointer durch die Queue geschleust wurden, auch vom empfangenden Prozeß durchgeführt werden kann. Andernfalls droht der Abschuß des Servers durch das Betriebssystem. Der Zugriff auf Speicher durch fremde Prozesse kann beim Anlegen des Speichers mittels bestimmter Optionen erfolgen.

Eine Stärke der Queues ist die Definition des Datenflusses innerhalb der Queue. IBM OS/2 Warp 4 stellt drei unterschiedliche Mechanismen für Queues zur Verfügung:

– *FIFO* (First in First out). Alle Daten werden vom Server in der Reihenfolge empfangen, wie sie auch von den Clients eingestellt wurden. Die erste gesendete Nachricht wird auch wieder als erste Nachricht empfangen.

– *LIFO* (Last in First out). Dieses stackorientierte Verfahren stellt die Reihenfolge auf den Kopf. Das zuletzt eingestellte Datum wird als erstes wieder empfangen. Die zuerst eingestellte Nachricht kann erst als letzte wieder ausgelesen werden.

– *Priority*. Jedes Element in der Queue kann mit einer Priorität versehen werden. Der Empfang der Daten seitens des Servers erfolgt dann entsprechend der Priorität.

Queues eignen sich besonders für die Abarbeitung von Aufgaben durch einen Server. Eine beliebige Anzahl von Clients teilt einem Server mit, was er tun soll. Dieser arbeitet dann diese Aufgaben in der Reihenfolge ab, die beim Anlegen der Queue vorgegeben wurde.

2.3.7 Semaphoren

Semaphoren kann man sich in etwa wie Signalgeber vorstellen. Zwei oder mehr Threads möchten ein kritische Aufgabe verrichten. Dies kann zum Beispiel der schreibende Zugriff auf eine gemeinsam genutzte Speicherstelle sein. Damit der Zugriff, und damit der Erfolg der Operation, nicht nach dem Zufallsprinzip erfolgt, wird eine Kontrollinstanz eingeschaltet.

Diese Kontrollinstanz wird beim Einstieg in die kritische Phase von jedem Beteiligten in Anspruch genommen. Alle mit dieser Aufgabe betrauten Threads begehren der Kontrollinstanz gegenüber den Wunsch auf Veränderung dieser Speicherstelle. Wenn dies nicht bereits von jemandem anderen in Anspruch genommen wurde, erhält der Anfrager den sofortigen Zugriff. Während dieser Phase ist der Zugriff für alle anderen interessierten Threads nicht möglich. Diese werden beim Äußern ihres Begehrens einfach blockiert. Auch hier handelt es sich um den bereits bekannten »Waiting for I/O«-Status. Wenn der Thread, der zuerst den Zugriff erhielt, seine Arbeit verrichtet hat, vermeldet er dieses der Kontrollinstanz und diese wird dem nächsten interessierten Thread den Zugriff erlauben.

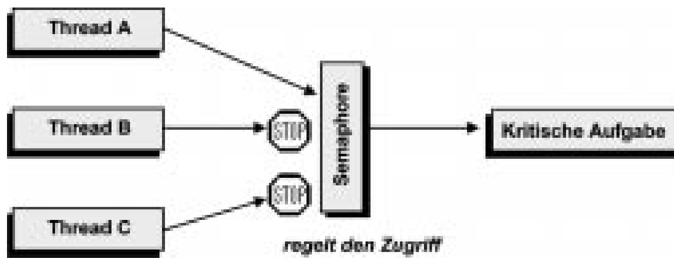


Abb. 2.6: Nur ein Thread erhält die Freigabe zu einer kritischen Aufgabe

Daraus wird ersichtlich, daß das System an dieser Stelle von der sauberen Programmierung und der Kooperation aller Partner abhängig ist. Ein nicht ordentlich zurückgesetzter Semaphor kann alle anderen Prozesse, die ebenfalls mit der gleichen Arbeit beschäftigt sind, blockieren. Erst nach Beendigung des schlecht programmierten Prozesses werden dann alle blockierten Ressourcen vom System freigegeben, und dazu würde dann der Semaphor gehören. Damit die Aufräumarbeiten nicht immer nur vom System vorgenommen werden müssen, kennen Semaphoren einen Timeout, der nach Ablauf dieser Zeit ohne eine Freigabe den Thread mit einem Fehlerhinweis aktiviert und so auf diesen Umstand hinweist. Leider gibt es aber auch einen undefinierten Timeout, der nie abläuft, und dieser wird fast von allen Anwendungen eingesetzt. Damit wird auch hier wieder ein sinnvoller Schutzmechanismus ad absurdum geführt.

OS/2 unterscheidet grundsätzlich zwischen drei Kategorien von Semaphoren:

Events. Event-Semaphoren werden für das Anzeigen eines Ereignisses eingesetzt. Ein oder mehrere Threads haben der Kontrollinstanz über einen Event-Semaphor angezeigt, daß sie auf ein bestimmtes Ereignis warten. Solange dieses Ereignis nicht auftritt, werden alle Threads angehalten.

Mutexe. Die Mutual-Exclusion-Semaphoren synchronisieren den Zugriff durch mehrere Prozesse oder Threads auf eine einzelne Aufgabe. Es handelt sich dabei, wie der Name schon sagt, um einen wechselseitigen Ausschluß. Wer zuerst angeklopft hat, ist auch zuerst an der Reihe. Wer zu spät kommt, erhält erst dann die Erlaubnis, mit der Arbeit fortzufahren, sobald der zur Zeit mit der Aufgabe betraute Prozeß oder Thread seine Arbeiten abgeschlossen hat, und den Semaphor wieder freigegeben hat.

MuxWaits. Diese Sonderform von Semaphor wird für sogenannte Semaphoren-Pools eingesetzt. Statt auf ein bestimmtes Signal oder die Freigabe einer bestimmten Speicherstelle zu warten, kann hier die Erlaubnis von einer Kette von Semaphoren abhängig gemacht werden. Solange nicht alle Semaphoren die Freigabe erteilt haben, werden die blockierten Threads nicht mit der Arbeit fortfahren können. MuxWait-Semaphoren können sowohl Mutex-Semaphoren-Ketten oder auch Event-Semaphoren-Ketten behandeln.

Innerhalb dieser drei Kategorien unterscheidet OS/2 Warp 4 noch zwischen zwei weiteren Arten von Semaphoren.

Global. Diese Semaphor-Kategorie ist für die Synchronisation von prozeßübergreifenden Threads gedacht. Das Anlegen dieses Semaphors geschieht mit Hilfe eines Devicenamens wie »\SEM\MeinSemaphor«. Alle Threads, denen dieser Semaphor bekannt ist, können einen Handle vom System erfragen und mit Hilfe dieses Handles wird dann das Freigeben und Blockieren von Threads gesteuert.

Privat. Semaphoren dieser Kategorie eignen sich nur für die Synchronisation von Threads eines Prozesses, da für die Synchronisation ein Datum aus dem Speicherkontext des Prozesses verwendet wird. Eine beliebige Variable wird hierfür als Signalgeber eingesetzt. Diese fungiert für alle an dem Semaphor interessierten Threads als Handle. Das Setzen dieser Variable blockiert alle anderen zugreifenden Threads. Das Zurücksetzen dieses Handles gibt das Semaphor wieder frei, und der nächste zur Zeit blockierte Thread wird wieder freigegeben. Private Semaphoren sind sehr schnell eingerichtet und gehören eigentlich in jedes gute OS/2-Multithreading-Programm.

Semaphoren werden dem Entwickler von OS/2-Anwendungen immer wieder über den Weg laufen, da die Synchronisation von gemeinsamen Aufgaben, durchgeführt durch unterschiedliche Threads, unter einem Multithreading-Betriebssystem zum Tagesgeschäft gehört. Obwohl relativ einfach anzuwenden, scheuen doch viele Umsteiger von den DOS-Plattformen, wo solche Anforderungen so gut wie nie vorkommen, den Einsatz dieser Hilfsmittel. Das Ergebnis sind dann meist Single-Threaded-Anwendungen, welche die zur Verfügung stehenden System-Ressourcen somit nicht ausreizen, oder schlicht und ergreifend Applikationen sind, die mangels geeigneter Synchronisation rein zufällig noch funktionieren.

Das Betriebssystem selbst und die meisten systemnahen Komponenten machen reichlich Gebrauch von Semaphoren. Sie sind schnell anzuwenden, kosten wenig Kraft und helfen bei der Entwicklung von effektiven, schnellen und sauberen Multitasking- und Multithreading-Anwendungen.

2.3.8 Shared Memory

Shared-Memory-Bereiche sind ebenfalls ein häufig anzutreffender Interprozeß-Kommunikationsweg unter OS/2. Vielfach wird Shared Memory selbst von den anderen Interprozeß-Kommunikationsmechanismen, wie zum Beispiel DDE, Drag and Drop oder Queues, unter der Hand durchgeführt, um die Aufgaben erfolgreich über Prozeßgrenzen hinweg durchführen zu können.

Ein Shared-Memory-Bereich ist ein Puffer, auf den zwei oder mehr Anwendungen Zugriff haben. Dabei kann dieser Bereich von den Applikationen beschrieben oder auch gelesen werden, je nachdem, wie dieser Speicherbereich definiert wurde. Zum Zugriff auf Shared-Memory-Bereiche wird von dem zugreifenden Prozeß ein Pointer auf die Daten benötigt. Danach ist der Zugriff freigegeben. Da der Zugriff auf fremde Speicherbereiche vom OS/2 per Definition verboten ist, müssen spezielle Vorkehrungen getroffen werden, damit dies möglich wird.

OS/2 unterscheidet zwischen zwei Arten von Shared-Memory-Bereichen. Es gibt »unbenannte Bereiche« und »benannte Bereiche«. Bei den benannten Speicherbereichen muß wieder mit Devicenamen gearbeitet werden. Ein Präfix »\SHAREMEM\« vor einem Speichernamen (zum Beispiel \SHAREMEM\MeinMemory) erlaubt die Aufnahme dieses Speichers in eine Shared-Memory-Device-Tabelle eines jeden OS/2-Rechners.

Der Zugriff auf Named Shared Memory ist denkbar einfach. Zum Öffnen wird lediglich der bekannte Name benötigt und schon ist der Zugriff erlaubt. Schon alleine die Tatsache, daß ein Prozeß den Namen eines Shared-Memory-Bereichs kennen muß, um darauf Zugriff zu erhalten, zeigt, daß dieser Kommunikationsweg nur den Prozessen möglich ist, die in der einen oder anderen Form miteinander bekannt sind.

Der Zugriff auf unbenannte Shared-Memory-Bereiche muß explizit vom Prozeß, der diesen Bereich angelegt hat, erfragt werden. Als Ergebnis einer positiven Rückmeldung wird lediglich der Zeiger auf diesen Bereich an den anfragenden Prozeß übermittelt. Mit diesem Zeiger erhält der anfragende Prozeß in der Regel gleichzeitig auch die Zugriffsrechte eingeräumt.

Der API-Befehl *DosAllocSharedMem* legt einen Shared-Memory-Bereich an. Mit Hilfe des API-Befehles *DosGetNamedSharedMem* wird der Zugriff auf einen solchen Bereich angefordert. Mit diesen API-Befehlen lassen sich sowohl unbenannte als auch benannte Shared-Memory-Bereiche verwalten. Bei Named-Shared-Memory-Bereichen muß lediglich der Device-Name angegeben werden, im Falle von Unnamed-Bereichen bleibt dieser einfach leer.

OS/2 unterscheidet zwischen zwei Methoden der Verwaltung von Shared-Memory-Bereichen:

– Zwei oder mehr Anwendungen benutzen den selben Shared-Memory-Block zur selben Zeit. Dabei wird das Lesen und Schreiben dieses Bereichs mit Hilfe von Semaphoren synchronisiert. Würde diese Synchronisation nicht durchgeführt, wäre auch hier wieder das Zufallsprinzip aktiv, das die Änderung des zufällig zuletzt schreibenden Prozesses vorziehen würde.

– Statt über Semaphoren den gemeinsam geteilten Speicherbereich zu schützen, benutzt eine Anwendung den Speicher, um ihre Daten zu füllen und übergibt erst dann den Speicher an eine andere Applikation, die diese Daten dann manipuliert. Dies ist das wohl häufigste Verfahren unter OS/2 Warp 4.

Eine Besonderheit muß noch erwähnt werden. Da Shared-Memory-Bereiche vom System nicht nur den beteiligten Prozessen zugeordnet werden, sondern allen Prozessen, landen diese in der Speicherzuordnung aller aktuell im System aktiven Prozesse, lediglich die beteiligten Prozesse erhalten den Zugriff auf diesen Speicher. Die Shared-Memory-Bereiche beginnen exakt an der 512-Mbyte-Obergrenze, die alle Prozesse als Limit haben. Mit jedem Shared-Memory-Objekt wandert dieser Bereich tiefer in Richtung der Applikationsdaten, die sich ja von unten nach oben im Speicher bewegen. Die Grenze dieser Ausbreitung ist die 448-Mbyte-Oberkante. Bis zu diesem Limit hinab können sich Shared-Memory-Bereiche ausweiten. Dies ist nicht viel, gerade mal eben 64 Mbyte. Es können in einem OS/2-Rechner zur Zeit maximal 64 Mbyte Shared-Memory-Bereiche existieren. Dieser Bereich wird aber auch den System-Ressourcen wie System-DLLs, die von allen Prozessen angezogen werden, zur Verfügung gestellt.

Es ist also nur noch eine Frage der Zeit, wann diese 448-Mbyte-Grenze gesprengt wird. Zu diesem Zeitpunkt wird dann wohl oder übel allen Prozessen der Prozeßrahmen von 512 Mbyte auf die theoretischen 4 Gbyte ausgeweitet werden müssen. Dies war zwar technisch schon immer möglich, wurde aber aus Gründen der Kompatibilität zu OS/2-1.x-Anwendungen aufrecht erhalten. Windows NT hat diese Grenzen auf Kosten der Kompatibilität, bereits zu einem sehr frühen Zeitpunkt fallen lassen, IBM hat das bei OS/2 ein wenig hinausgezögert.

2.4 Speichermanagement und Speicherschutz

Dem Betriebssystem IBM OS/2 Warp 4 muß als Multitasking- und Multithreading-fähiges Operating System ein solides Grundgerüst für die Verwaltung des zur Verfügung stehenden Hauptspeichers zur Seite gestellt werden.

Dies umfaßt zum einen das schnelle und effektive Verwalten des verfügbaren Speichers, auf der anderen Seite aber auch den wirksamen Schutz des Speichers gegenüber nicht autorisierten Zugriffen.

2.4.1 Speichermanagement

Seit OS/2 Version 2 handelt es sich bei OS/2 weitestgehend um ein 32-bittiges Betriebssystem, das den modernen CPUs direkt auf den Leib geschnitten ist. Mit dem Intel-Prozessor 80386 stand die erste dieser 32-Bit-CPUs der Version 2 von OS/2 zur Verfügung.

Eines der elementaren Neuerungen war unter anderem die Einführung eines sogenannten Flat-Memory-Modells. OS/2-Versionen vor 2.0 waren noch auf ein segmentiertes Speichermodell aufgebaut.

Das segmentierte Speichermodell des Intel-Prozessors 80286, der als Grundstein für die früheren OS/2-Versionen verwendet wurde, erlaubte den Zugriff auf 16 Mbyte physikalischen Hauptspeicher. Dies war schon eine drastische Erweiterung gegenüber der herkömmlichen 640-Kbyte-Grenze bei DOS, aber immer noch nicht genug. Schnell war klar, daß die 16-bittige Auslegung des Intel 80286 und der OS/2-Versionen 1.x nicht lange überleben würden. 32 Bit war schon frühzeitig das Ziel.

Das segmentierte Speichermodell galt als 16:16-Modell, was bedeutet, daß ein 16-Bit-Selektor (Basiszeiger), kombiniert mit einem 16-Bit-Offset, auf eine bestimmte Speicherstelle zeigte. Dieses Modell hatte allerdings einen gewaltigen Nachteil. Speichersegmente konnten fast beliebige Größen annehmen, vor allen Dingen aber sehr klein werden. Die Verwaltung von diesen Kuchenstücken beliebiger Größe barg doch einige Fallen. So mußte der Speicher-Manager die Segmente im physikalisch zur Verfügung stehenden Speicher hin- und herschieben, um eine optimale Nutzung zu erreichen, denn nur zusammenhängender Speicherbereich konnte als Segment einer Anwendung übergeben werden. Man muß sich nur einmal vor Augen halten, mit wieviel Arbeit der Memory-Manager beschäftigt war, nur weil das System keinen besseren Weg zur Verfügung stellte. In Systemen, auf denen sehr viel Last zu verzeichnen war, wie zum Beispiel der LAN Server oder Datenbank-Server-Umgebungen, spielte das System schier verrückt. Jede kurze Denkpause wurde unter anderem mit der Reorganisation des Speichers verbracht.

Das Flat-Memory-Modell ab der OS/2-Version 2 geht einen komplett anderen Weg. Es wird als 0:32-Modell bezeichnet, da kein Basiszeiger mehr benötigt wird, sondern der Offset mit seiner 32-Bit-Größe direkt jede beliebige Speicherstelle adressieren kann.

Nun, das klingt alles sehr einfach, aber damit es nicht zu einfach wurde, mußten zusätzlich einige weitere Features eingebaut werden. Diese werden bereits von der Hardware, sprich der CPU, unterstützt. Was lag da also näher, als diese Eigenschaften direkt mit einzusetzen?

Die Intel-Prozessoren neuerer Bauart kennen den Begriff des Paging. Eine Page, eine Seite, ist zur Zeit exakt 4 Kbyte groß. Dieses Paging erleichtert nun die Verwaltung von physikalischem und virtuellem Speicher ungemein.

Existiert in einem Rechner ein physikalischer Hauptspeicher von 32 Mbyte, so wird dieser in gleich große Stücke von 4 Kbyte aufgeteilt. Auf der anderen Seite verwaltet das System einen theoretisch zur Verfügung stehenden Speicher von 4 Gbyte ebenfalls schön in 4 Kbyte große Stücke aufgeteilt.

Fordert ein Programm Speicher an, so wird dies auf eine 4-Kbyte-Seite ausgerichtet und im virtuellen Speicher als besetzt gekennzeichnet. Gleichzeitig wird, wenn vorhanden, diese Seite im physikalischen Hauptspeicher abgelegt. Die Verknüpfung der Seite zwischen virtuellem Adressraum und physikalischem Speicher erfolgt durch Seitenverzeichnisse, die jedem Prozeß zugeordnet sind, sowie Seitentabellen. Einfach dargestellt, schaut das System bei einem Zugriff auf den virtuellen Speicher nach, wo diese Seite denn nun tatsächlich abgelegt wurde.

Der theoretisch zur Verfügung stehende Hauptspeicher von 4 Gbyte läßt sich an Hand der Seitenverzeichnisse und Seitentabellen nachvollziehen. Das Seitenverzeichnis selbst ist eine Seitentabelle. Jede Seitentabelle kann maximal 1024 Pointer von einer Größe von 32 Bit aufnehmen. Also kann das Seitenverzeichnis auf maximal 1024 Seitentabellen zeigen, die wiederum auf 1024 Seiten zeigen können. Da jede Seite 4 Kbyte groß sein kann, entspricht dies dem maximal möglichen virtuellen Adreßraum von 1024 * 1024 * 4096 Byte = 4 Gbyte.

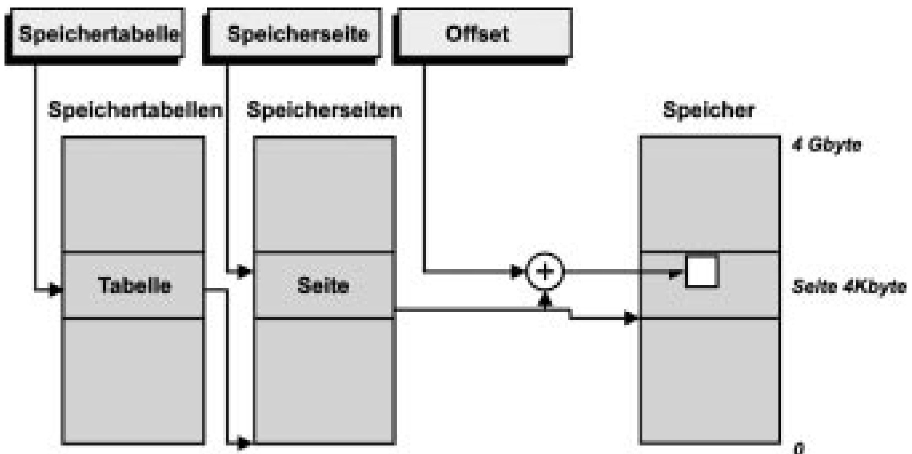


Abb. 2.7: Der Zugriff auf eine Speicherstelle erfolgt in drei Stufen mittels Zeiger auf Speichertabellen, Speicherseite und Speicherzelle

Ist der zur Verfügung stehende Hauptspeicher belegt, so wird die zuletzt genutzte Seite einfach ausgelagert und an deren Stelle die neue Seite eingetragen. IBM OS/2 Warp 4 unterscheidet vier verschiedene Arten der Auslagerung:

- *Discard*. Eine Seite wird dann komplett aus dem System entfernt, wenn sie ohne großen Aufwand wieder aus anderer Quelle besorgt werden kann. So gehören in der Regel ausführbare Programme sowie DLLs zu den Kandidaten, deren Code aus dem System entfernt wird, wenn es einmal etwas enger werden sollte. Wird der Code wieder benötigt, so wird er aus der entsprechenden Datei seitenweise wieder nachgeladen.
- *Fixed*. Eine fest zugeordnete Seite kann nicht ausgelagert werden. Hierbei handelt es sich um Speicher, der dem Kernel zugeordnet ist. Dieser Art von Speicher verbleibt immer im Hauptspeicher.

– *Preloaded*. Diese Sonderform der Auslagerung kam erst mit OS/2 Warp 3 richtig zum Tragen. Auf Grund neuer Packalgorithmen für EXE- und DLL-Module (EXEPACK2) war es nicht mehr möglich, eine Seite direkt aus einem solchen Modul nachzuladen, ohne dieses zuvor zu entpacken. Dies benötigt zu viel Systemzeit. Aus diesem Grund erhalten die Systemdateien, von denen erwartet wird, daß sie immer benötigt werden, auf der anderen Seite aber eigentlich discardable sind, eine spezielle Behandlung. Preloaded-Module werden beim ersten Zugriff komplett in den Hauptspeicher geladen. Bei den Systemdateien werden die Seiten dieser Module direkt in der Swapper.Dat hinterlegt. So können die Seiten solcher Module auch entfernt werden, müssen aber nicht jedesmal in die Swapper.Dat kopiert werden. Das Nachladen aus der Originaldatei mit einem notwendigen Entpackungsprozeß entfällt auf diese Weise ebenso. Der Vorteil liegt auf der Hand. Beim Laden dieser speziellen Systemdateien wird das Modul entpackt, in kleine 4-Kbyte-Stücke aufgeteilt und unter anderem in die Swapper.Dat kopiert. Das Nachladen von 4-Kbyte-Seiten geht sehr schnell vonstatten. So mancher Benutzer von OS/2 wird zwar sofort nervös, wenn die SWAPPER.DAT bereits nach dem Starten des Betriebssystems mit Inhalten gefüllt ist, dies hat aber ausschließlich Performance-Gründe.

– *Swappable*. Eine austauschbare Speicherseite wird bei Systemengpässen in eine Auslagerungsdatei mit dem Namen Swapper.Dat ausgelagert. Eine solche Seite muß nicht permanent im Speicher verbleiben, kann aber andererseits nicht aus einer anderen Quelle rekonstruiert werden. Es handelt sich hierbei fast ausschließlich um die Daten von Programmen.

Wird auf eine ausgelagerte Seite zugegriffen, so erkennt das System dies an einem speziellen Flag. Das System löst eine »Pagemiss-Exception« aus, die die betreffende Seite aus der Swapper.Dat holt oder aus dem Originalmodul nachlädt. Diese Seite wird dann in den physikalischen Speicher eingeblendet und das Programm kann anschließend fortfahren, als wäre nichts geschehen. Sollte bei dieser Aktion der physikalische Speicher gefüllt sein, so muß natürlich vorher die älteste Seite ausgelagert werden. Dies hört sich schlimmer an als es ist. Die CPU unterstützt diesen Mechanismus, wo es nur geht, und da die Swapper.Dat-Datei permanent geöffnet ist, muß nicht mit großen Verzögerungen gerechnet werden. Mehr Hauptspeicher ist aber auf alle Fälle die bessere Lösung.

Interessant an diesem seitenorientierten Design ist die Tatsache, daß es vollkommen unerheblich ist, an welcher Stelle im physikalischen Speicher die Seiten existieren. Ein Programm, das normalerweise von oben nach unten abgearbeitet wird, findet sich auf einmal in Seiten zerhackt an verschiedenen Stellen im System wieder. Das Programm und der Benutzer merken von dieser Aufteilung nichts. Der Vorteil gegenüber dem segmentierten Modell ist unter anderem, daß das System den physikalischen Hauptspeicher nicht mehr reorganisieren muß. Jede Seite ist exakt 4 Kbyte groß und paßt an jede Stelle in dieses Raster hinein. Als kleiner Nachteil darf nicht unerwähnt bleiben, daß 4 Kbyte natürlich auch die kleinste Größe ist, die das System zur Verfügung stellen kann. Durch sogenannte »Sub-Allokationen« des Systems und durch intelligente Speichermanager in allen Hochsprachen wird dieses Problem aber umgangen. Diese Speichermanager allokieren einfach auf Verdacht eine genügend große Menge Memory und geben diesen in kleinen Happen weiter. Beim IBM-VisualAge-C++-Compiler werden so als kleinste Speichereinheit 32 Byte ermöglicht.

Wie bereits im Kapitel »Sessions, Prozesse und Threads« aufgeführt, hat nur das Betriebssystem selbst Zugriff auf den gesamten virtuellen Adressraum. Hierzu wird die Privilegebene 0 benötigt. Den Anwendungsprozessen hingegen wird lediglich ein maximal verfügbarer virtueller Hauptspeicher von 512 Mbyte bereitgestellt. Von diesen gehen aber 64 Mbyte für Shared-Ressourcen ab, so daß jedem Prozeß lediglich maximal 448 Mbyte zur Verfügung stehen. Ein Zugriff eines Anwendungsprogramms auf den Adressraum jenseits der 512 Mbyte wird grundsätzlich durch die entsprechende GPF-Exception (General Protection Fault) abgelehnt und mit einem Abschuß des Prozesses begleitet.

2.4.2 Speicherschutz

Das unter OS/2 eingesetzte Speicherschutzverfahren entspricht heute nur zu einem geringen Teil den tatsächlichen Möglichkeiten, die von den Intel-Prozessoren bereitgestellt werden.

Zunächst einmal unterscheidet der Prozessor zwischen vier Privilegebenen, auch Ringe genannt. Ring 0 besitzt dabei den höchsten Schutz und Ring 3 den niedrigsten Schutz. Betriebssystemteile werden üblicherweise auf Ring 0 zu finden sein, während Anwendungsprogramme auf Ring 3 eingeordnet werden. Jeder Ring beschreibt die Rechte, die der Code auf dieser Ebene hat. Werden die Rechte überschritten, so erfolgen wiederum entsprechende Exceptions.

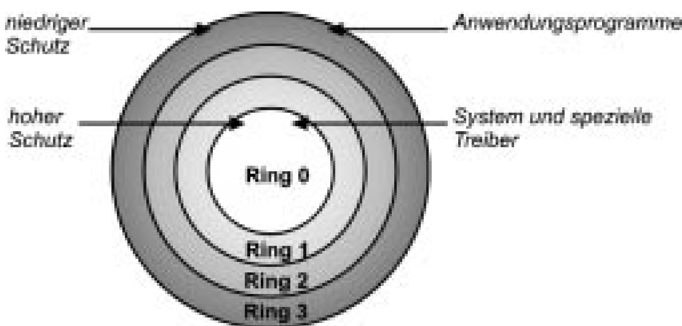


Abb. 2.8: Die vier Ebenen des Speicherschutzes unter OS/2

Der Zugriff auf höherwertige Schutzräume wird vom System über sogenannte »Call-Gates« ermöglicht. So sind viele API-Aufrufe nichts anderes als ein Einstieg in einen niedrigeren Ring. Die vom Anwendungsprogramm gewünschte Funktionalität wird – nach Parameterprüfung – huckepack von der API-Funktion in einen Schutzraum getragen und dort unter Aufsicht ausgeführt. Nach Abschluß der Aufgabe geleitet der API-Aufruf diesen Funktionsaufruf wieder sicher aus dem Schutzraum hinaus. Dies hat natürlich nur Vorteile. Jeder Aufruf des Anwendungsprogramms erfolgt mehr oder weniger unter der Kontrolle des Systems, und beim Einschleusen in die Schutzräume können Kernel-Funktionen ausgeführt werden, die auf Ring 3 nicht möglich sind. Bei diesem zwangsläufig durchzuführenden Wechsel von dem »User Mode« in den »Protected Mode«, in dem erst der Zugriff auf den gesamten virtuellen Hauptspeicher möglich ist, geht natürlich etwas Prozessorzeit verloren, denn ein solcher

Wechsel erzwingt eine Umschaltung der CPU. Aus diesem Grund sind manche Betriebssystem wie z.B. Windows dazu übergegangen, Teile des Betriebssystems und die Anwendungsprogramme auf einem Ring ablaufen zu lassen. Dies bringt zwar eine etwas gesteigerte Programmausführung, hat aber den Nachteil, daß nun das Anwendungsprogramm im Fehlerfall diesen gleichberechtigten Betriebssystemteil direkt mit abschiessen kann. In einem richtiges Protected-System sollten die Privilegringe hingegen vollständig ausgeschöpft werden.

Jeder Prozeß besitzt unter 32-Bit-OS/2 einen eigenen Satz von Seitentabellen, die auf den virtuellen Speicher zeigen. Wechselt das Betriebssystem den aktiven Prozeß, so werden die dem Prozeß zugeordneten Seitentabellen ebenfalls ausgetauscht. Der Scheduler hat sich also nicht nur mit einem Thread-Kontext (zum Beispiel CPU-Register) zu beschäftigen, er muß gleichzeitig auch den Prozeß-Kontext austauschen. Dieser Prozeß-Kontext ist aber nun die Basis für den Speicherschutz auf Anwendungsebene, denn diese einem Prozeß zugeordneten Seitentabellen sind eindeutig und geschützt. Sie erinnern sich, jeder Thread eines Prozesses erbt die Ressourcen und die Rechte dieses Prozesses.

Vereinfacht gesagt, greift ein Anwendungsprogramm auf Speicher zu, und ist dieser nicht in seinen eigenen Seitentabellen zu finden, so ist dies zunächst ein Fehler. Nun kann aber eingewendet werden, daß es Shared-Ressourcen gibt, und daß ein Prozeß einem anderen Prozeß Rechte an gemeinsam genutzten Ressourcen einräumen kann. Dies ist korrekt, ändert aber nichts an der Überprüfung des Zugriffs. Erhält ein Prozeß Shared-Ressourcen, von wem auch immer, so werden die betreffenden Seiten in die Seitentabelle des Prozesses eingetragen, der diese Rechte erhält. Es ist unter OS/2 gängige Praxis, daß eine Seite gleichzeitig in den Seitentabellen der unterschiedlichsten Prozesse enthalten ist. Denken Sie nur an gemeinsam genutzte DLLs. Diese sind ebenfalls in allen Prozessen eingetragen, die auf diese DLL zugreifen.

Jenseits der globalen Überprüfung, ob eine Seite in der Prozeß »Seitentabelle« eingetragen ist oder nicht, muß das System noch weitere Prüfungen vornehmen. Jede Seite kann ein Vielzahl verschiedener Stati haben:

- *READ*. Der Zugriff auf diese Seite ist nur lesend möglich. Jeder Schreibzugriff wird abgeblockt. Das READ-Attribut können ausführbare Programmseiten erhalten (siehe EXECUTE), kann aber auch sonst anderweitig verwendet werden.
- *WRITE*. Der Schreibzugriff auf diese Seite ist möglich. Datenbereiche von Programmen gehören üblicherweise in diese Attribut-Kategorie. Bei 80386-Prozessoren bedeutet der WRITE-Zugriff gleichzeitig auch READ- und EXECUTE-Recht.
- *EXECUTE*. Der Code in dieser Seite kann ausgeführt werden. OS/2 kennzeichnet Programmcode von EXE- und DLL-Modulen mit diesem Attribut, nutzt dieses darüberhinaus aber nicht weiter aus.
- *GUARD*. Diese Seite ist im System als allokiert gekennzeichnet, aber noch nicht physikalisch realisiert. Jeder READ/WRITE- oder auch EXECUTE-Zugriff auf eine Guard-Page generiert einen Page-Fault. Der standardmäßig vom System installierte Exception-Handler versucht, aus der Guard-Page eine COMMIT-Page zu machen und versucht dann, die nächste Seite als Guard-Page zu kennzeichnen. Da dieser Exception-Handler überschrieben werden kann, können Guard-Pages auch zum applikationsspezifischen Ausdehnen von Speicherbereichen eingesetzt werden.

– *COMMITED*. Diese Seite ist physikalisch realisiert. Der Zugriff im Rahmen der Rechte ist direkt möglich. Eine COMMITED-Seite hatte normalerweise noch weitere beschreibende Attribute wie READ, WRITE oder EXECUTE.

Natürlich sind Kombinationen dieser Stati für eine Seite möglich. So kann eine Seite READ/WRITE gesetzt haben, oder auch READ/WRITE/GUARD.

Wurde über die Seitentabelle des Prozesses die Seite des Zugriffs gefunden, muß der Status dieser Seite noch überprüft werden. Versucht das Anwendungsprogramm den schreibenden Zugriff auf eine Seite, die nur den Lesezugriff erlaubt, so wird dieser Zugriff ebenfalls abgelehnt.

2.5 Dateisysteme

Bereits seit Version 1.2 verfügt das Betriebssystem OS/2 über ein sogenannten installierbares Dateisystem. Das bedeutet, daß das System nicht mehr fest mit dem »alten« FAT-(File Allocation Table) Dateisystem verdrahtet ist, sondern neuere Dateisysteme unterstützt.

Ein Dateisystem stellt alle Mechanismen zur Verfügung, um dem System und den Anwendungsprogrammen den Zugriff auf Verzeichnisse und Dateien zu ermöglichen. Nun können aber mehrere sogenannte Devices gleichzeitig in einem System existieren. Es gibt da zum Beispiel neben den bekannten Festplatten- und CD-ROM-Laufwerken auch exotischere Varianten wie WORM (Write Once Read Many) oder die aktuellen Zwitter, bestehend aus CD-ROM-Laufwerk und MO-(Magneto Optischen) Laufwerken.

Oftmals kommt mit einem neuen Device gleichzeitig eine neue Zugriffsart auf dieses Device. Damit diese nahtlos in OS/2 eingebunden werden können, liegt eine Schicht zwischen der Anforderung an ein Dateisystem und der tatsächlichen Ausführung dieser Aktion.

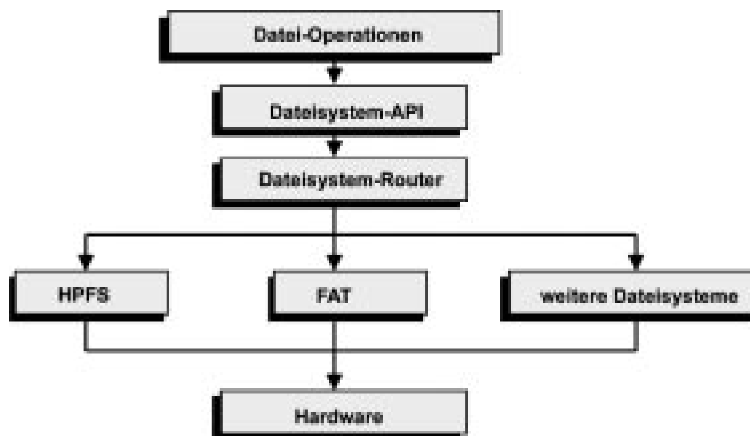


Abb. 2.9: Verteilung der Dateioperationen auf installierte Dateisysteme

Wird vom Anwendungsprogramm eine Dateioperation durchgeführt, so geschieht dies mit Hilfe der entsprechenden API-Befehle des Kernels. Bestandteil jeder Anforderung ist ein Devicename (COM:, LPT:), ein logischer Laufwerksbuchstabe (C:, D:) oder eine Netzwerkadresse (\\MYCOMPUTER). Mit Hilfe dieser Angaben stellt ein eigenständiger Manager im System fest, welches Dateisystem für die Bearbeitung dieser Anfrage notwendig ist. Von diesem Manager wird die Anforderung an das zutreffende System weitergeleitet und dann dort zur Ausführung gebracht.

Jedes Dateisystem kann seine eigene Verwaltung in OS/2 einbinden. Das System stellt hier keine Maßgaben. Traditionell wird bei Dateisystemen mit Dateien und Verzeichnissen gearbeitet, dies muß aber nicht sein. Jedenfalls erlaubt das installierbare Dateisystem Dateiobjekte jeglicher Art. Eine Datei enthält normalerweise einfach nur Daten, während ein Verzeichnis Dateien sowie weitere Verzeichnisse enthalten kann. Von OS/2 nicht unterstützt wird der »Link-Mechanismus«, der das gleichzeitige Vorhandensein einer einzigen Datei in mehreren Verzeichnissen erlaubt. Eine Änderung dieser Datei in einem Verzeichnis würde immer das Original verändern und somit alle auf dieses Original verweisende gelinkten Dateien ebenfalls. Dieser Mechanismus wird gerade von Entwicklern schmerzlich vermißt.

Der Link-Mechanismus wird allerdings von der Workplace Shell benötigt (Shadow-bzw. Schattenobjekte). Da OS/2 diese Unterstützung nicht vom Dateisystem her bietet, besitzt die WPS einen eigenen Mechanismus, der dies umgeht. Für den Entwickler oder Benutzer, der diese Funktionalität benötigt, gibt es ein sogenanntes IBM-EWS-(Employee Written Software) Programm. Das Utility trägt den Namen TVFS (Toronto Virtual File System).

Damit es keine unliebsamen Überraschungen gibt, werden die Dateinamen immer nur eindeutig in jedem System erlaubt. Damit dies auch so bleibt, gibt es eine eigenständige Instanz, die dies untersucht. Jeder Name eines Dateisystems wird in einer Namenstabelle gehalten, um die Eindeutigkeit eines neuen Dateiobjekts schnell feststellen zu können.

Allen Dateisystemen eigen sind die sogenannten symbolischen Platzhalter:

- * steht für eine beliebige Anzahl Buchstaben in einem Dateinamensteil
- ? steht für exakt einen Buchstaben in einem Dateinamensteil
- \ steht für das Wurzelverzeichnis auf einem Laufwerk, wird aber auch gleichzeitig als Trennzeichen zwischen Verzeichnisnamen und Verzeichnis- und Dateinamen verwendet
- .. zeigt auf das Elternverzeichnis
- . zeigt auf das aktuelle Verzeichnis

Einige Beispiele:

- D:\ = Wurzelverzeichnis auf Laufwerk D
- ..\. = alle Dateien im Elternverzeichnis

»?UGO.B*« = alle Dateien und Verzeichnisse, die mit einem beliebigen Buchstaben beginnen, gefolgt von dem festen Text »UGO.B«, dem wiederum ein beliebiger Text folgen kann

Bei Festplatten-Laufwerken wird zwischen Platten, Köpfen, Tracks und Sektoren unterschieden. Es können mehrere Platten in einem Laufwerk enthalten sein. Eine Platte kann bis zu 2 Schreib-/Leseköpfe besitzen. Auf jeder Plattenseite gibt es eine Anzahl von Tracks (kreisförmige Spuren), die wiederum in eine Anzahl von Sektoren (Abschnitte) unterteilt sind. Dieses System kann natürlich bei anderen Devices ganz anders aussehen, das installierbare Dateisystem hilft über diese Hürde.

Der Zugriff auf Dateioperationen erfolgt ausschließlich mittels des File-System-APIs. Der Befehl *DosOpen* öffnet zum Beispiel eine Datei und liefert dem Anwendungsprogramm einen Handle (eine eindeutige Nummer). Alle weiteren Operationen mit dieser Datei werden vom Programm mit Hilfe dieser Nummer durchgeführt. Mit *DosRead* werden Daten gelesen, mit *DosWrite* werden diese wiederum geschrieben und die Datei wird zum Schluß mit *DosClose* wieder geschlossen. Weitere APIs sind für die Verwaltung der Attribute (Schreibschutz, Archiv etc.), erweiterten Attribute (.TYPE, .COMMENT), Locking von Dateibereichen sowie mit der Positionierung des Dateizeigers innerhalb einer Datei zuständig.

Der Zugriff auf Dateien erlaubt die Angabe eines sogenannten Dateiöffnungs-Modus. Wird eine Datei von einem Programm geöffnet, so kann dieses angeben, ob die Datei für weitere Anwendung gesperrt bleibt, oder ob andere Anwendungen gleichzeitig mit dieser Datei arbeiten können. Weiterhin kann angegeben werden, ob diese Datei im Lese-, Schreib- oder Lese- und Schreibmodus geöffnet wird. Eine Datei, die sich im Lesemodus befindet, kann nicht beschrieben werden. Kombinationen dieser Dateiöffnungs-Modi sind erlaubt und in den meisten Fällen sogar erforderlich. So kann von einem Programm angegeben werden, daß es diese Datei im Schreib-/Lesemodus öffnet und gleichzeitig den Zugriff auf diese Datei durch andere Anwendungen verbietet. Man spricht in diesem speziellen Fall auch von exklusivem Zugriff.

Zur Zeit unterstützt OS/2 zwei lokale Dateisysteme:

– FAT (File Allocation Table)

– HPFS (High Performance File System)

In den folgenden Kapiteln werden wir uns diese beiden Dateisysteme ein wenig näher betrachten.

2.5.1 FAT (File Allocation Table)

Das FAT-Dateisystem ist bereits sehr alt, bekannt wurde es durch DOS.

FAT erlaubt die Verwaltung von Dateien und Verzeichnissen nach der sogenannten 8.3-(eight dot three) Konvention. Jedes Dateiobjekt kann sich innerhalb dieses schmalen Spielraums bewegen. So ist zum Beispiel folgende qualifizierte Dateiangabe erlaubt:

D:\MYDIR.OLD\MYDIR\MYFILE.DAT

Auf dem Laufwerk D: gibt es ein Hauptverzeichnis mit dem Namen Mydir.Old. Dieses wiederum enthält ein Unterverzeichnis mit dem Namen Mydir und dort findet sich eine Datei mit dem Namen »Myfile.Dat. Die Bezeichner »MYDIR» sowie »MYFILE« können aus nicht

mehr als 8 Zeichen bestehen, die Erweiterungen »OLD« sowie »DAT« sind auf eben diese drei Stellen beschränkt.

Jeder Teil einer Pfadangabe oder eines Dateinamens ist auf eben diese 8.3-Namensgebung begrenzt. In Zeiten, in denen sich auf einem durchschnittlichen Computer bereits zehntausende Dateien befinden können, wird diese 8.3-Konvention zu einem Korsett, das letztendlich mehr Probleme schafft, als es lieb sein kann.

FAT wurde im Laufe der Zeit immer weiter optimiert und erlaubt unter OS/2 auch den Zugriff auf Partitionen, die größer als 32 Mbyte sind. Dies war unter DOS das alte Limit.

Physikalisch existiert auf jedem FAT-Laufwerk eine 16-Bit-Namenstabelle, die sich wie ein Stream aus Dateinamen darstellt. Für die Speicherung dieser Namen ist nur begrenzter Platz vorhanden. Wird dieser ausgeschöpft, so muß das System einen weiteren Platz anfordern, der dann beliebig auf dem Laufwerk positioniert werden kann. Die Verknüpfung dieser beiden Bereichen werden dann mit Hilfe von Zeigern aufgebaut.

Eines der großen Probleme mit FAT ist seine Tendenz, sehr schnell zu fragmentieren. Eine Datei unter FAT wird zerstückelt in logische Einheiten, die dann physikalisch in Sektoren abgelegt werden. Diese Kuchenstücke können sich allerdings auf der Festplatte verstreut befinden, nämlich dann, wenn die Datei nicht komplett an einem Stück abgespeichert werden kann. In diesem Fall werden einfach die nächsten freien Sektoren verwendet, um die einzelnen Teile einer Datei abspeichern zu können. Je öfter dies geschieht, und im normalen Betrieb passiert dies sehr oft, um so verstreuter werden die Dateibestandteile. Das Lesen einer so verstreuten Datei wird für das System immer aufwendiger. Die Position der Köpfe muß immer wieder angepaßt werden, bis alle Teile dieser Datei gelesen sind. Das kostet wertvolle Zeit. Spezielle Defragmentier-Programme können dieses Problem beheben, dies kann aber nur dann geschehen, wenn kein anderes Programm auf die Platte zugreift. In einer Dateiserver-Umgebung ist dies nahezu unmöglich.

Der Umgang mit erweiterten Dateiattributen ist FAT von Hause aus fremd. Damit OS/2 seine erweiterten Attribute auf einem FAT-Dateisystem abbilden kann, wurde eine »Krücke« ins Leben gerufen. OS/2 reserviert spezielle Sektoren auf der jeweiligen Festplatten-Partition und markiert diese im Verzeichnis als besetzt. In diesem Block werden alle erweiterten Dateiattribute dieser Partition gespeichert. Auch dieses Verfahren hat einen gewaltigen Nachteil. Zum Kopieren einer Datei mit erweiterten Dateiattributen muß nicht nur die Datei aus seinen fragmentierten Bestandteilen wieder zusammengesetzt werden, sondern die erweiterten Dateiattribute müssen im betreffenden Sektorenbereich gesucht und der zu kopierenden Datei beigelegt werden. Auch dies ist sehr zeitintensiv.

Da das OS/2-FAT mit den langen Dateinamen des Systems umgehen können muß, wurde auch hier ein Umweg eingebaut. Lange Dateinamen werden vom System in einen eindeutigen Namen umgewandelt, der der 8.3-Konvention entspricht. Der Originalname wird dafür in einem EA mit der Bezeichnung ».LONGNAME« abgespeichert. Wird auf einem FAT-System nach einem Dateinamen gesucht, der nicht der 8.3-Konvention entspricht, so muß mit Hilfe der .LONGNAME-EA-Einträge zunächst der passende 8.3-konforme Name ermittelt werden. Was dieser Vorgang an Kraft kosten kann, muß nicht erläutert werden.

Seit OS/2 Version 2 besitzt das FAT-System einen eigenen Caching-Mechanismus. Ein Cache erlaubt die temporäre Aufnahme von Dateibestandteilen in einen speziellen Bereich des Hauptspeichers. Wird zufällig der selbe Dateibestandteil nochmals benötigt, so kann diese Anforderung ohne Zugriff auf das langsame Festplattenmedium direkt aus dem Hauptspeicher befriedigt werden. Ist der angeforderte Bereich nicht im Cache enthalten, so wird dessen Inhalt verworfen und während der Beschaffung der angeforderten Daten gleichzeitig aufgefrischt. In einer solchen Umgebung ist es wichtig, daß der angeforderte Dateiinhalt so oft wie möglich im Cache enthalten ist. Intelligente Cache-Systeme verfügen daher um ausgefeilte Mechanismen, mit denen dies bewerkstelligt wird.

Weitere Mechanismen wie LAZY WRITE, das Schreiben der Daten zu einem Zeitpunkt, wenn das System Zeit hierzu hat, sind da schon von einem anderen Kaliber. Bei einem Absturz des Systems können Daten verloren gehen. Deshalb lohnen sich solche Mechanismen nur in wirklich stabilen Systemen, alles andere ist fahrlässig. Auf der anderen Seite profitiert ein gut funktionierendes System sehr stark von solchen Optimierungsmöglichkeiten.

2.5.2 HPFS (High Performance File System)

Das FAT-System war sehr gut für das DOS und seine Nachfolger geeignet, solange es sich auf Einzelplatzsystemen mit wenig Multitasking bewegte. Sobald die Last des Systems anzieht, kann FAT ganz schön in die Knie gehen. Die Antwort darauf ist HPFS.

HPFS wurde von Microsoft während der Zusammenarbeit von IBM und Microsoft an OS/2 Version 1.x entwickelt. HPFS hebt nahezu jedes Limit der FAT-Dateiverwaltung auf.

HPFS formatiert eine Partition nach ganz anderen Gesichtspunkten als FAT. Das Verzeichnis wird nicht am Anfang der Partition gespeichert, sondern in der Mitte, um den durchschnittlichen Weg des Schreib-/Lesekopfes zu minimieren. Die Partition wird in 8 Mbyte große Bands (Bänder) aufgeteilt, in denen Platz für die Dateien geschaffen wird. Die erweiterten Dateiattribute der Dateien befinden sich direkt bei der Datei, oder wenn die EAs 64 Kbyte überschreiten, direkt in der Nähe. Die Verzeichnisse werden in »B-Bäumen« organisiert, um den blitzschnellen Zugriff auf eine Datei mit Hilfe eines Namens gewährleisten zu können.

Apropos Namen: HPFS hebt die 8.3-Konvention des FAT auf und erlaubt Dateinamen, die bis zu 254 Zeichen lang sein können. Eine voll qualifizierte Verzeichnis- und Dateinamensangabe kann bis zu 261 Zeichen beinhalten. Die Groß- und Kleinschreibung von Dateinamen und Verzeichnissen wird beibehalten. Das bedeutet, daß der Dateiname immer in Groß- und Kleinschrift gespeichert und auch wieder angezeigt wird. Bei einer Suche nach Dateinamen wird glücklicherweise aber nicht die Groß- und Kleinschrift unterschieden. Auf diese Weise werden endlich aussagekräftige Dateinamen wie der folgende möglich:

»D:\Dokumente\Firma XYZ\Beschwerde über mangelnde Unterstützung.DOK«

Beachten Sie bitte die Leerzeichen im Verzeichnis- und im Dateinamen. Diese werden unter HPFS unterstützt.

Die Dateinamenserweiterung DOK könnte entfallen, da die Zuordnung einer Datei zu einer bestimmten Anwendung nun auch über erweiterte Dateiattribute aufgebaut werden kann, da

aber noch nicht alle Anwendungen mit diesen arbeiten, bietet sich der Einsatz von Dateinamenserweiterungen nach wie vor an.

HPFS ist in vielen Operationen performancemäßig dem FAT überlegen. Die Fragmentierung wird zwar nicht ganz vermieden, aber mit Hilfe der geänderten Struktur der Dateiobjekte minimiert. Geringere Fragmentierung kommt gleichzeitig einem schnelleren Zugriff auf eine Datei gleich.

HPFS kommt mit allen bekannten Cache-Arten und erlaubt so die Einstellung des Systems nach den tatsächlichen Gegebenheiten. Eine solch umfangreiche Dateiverwaltung kann man natürlich nicht zum Nulltarif erwerben. Das HPFS-Verwaltungsprogramm belegt mit seinem Code und seinen Daten über 500 Kbyte Hauptspeicher. Der Einsatz ist also nur empfehlenswert in Umgebungen, in denen nicht ohnehin schon zuwenig Hauptspeicher vorhanden ist. Bei Partitionen unter 80 Mbyte lohnt sich der Einsatz von HPFS ohnehin nicht. Bei allen anderen Fällen ist der Einsatz von HPFS dem des FAT vorzuziehen.

Eines muß allerdings festgestellt werden: Das im OS/2 integrierte DOS und das integrierte WINOS2 kann auf HPFS formatierte Festplatten zugreifen, inklusive erweiterten Dateiattributen und langen Dateinamen. Dies trifft aber nicht auf reines DOS zu. Muß der Zugriff von DOS auf die Daten gewährleistet bleiben, so hilft nur die Formatierung der Partition mit FAT. Es gibt zwar auch hierfür eine kleine Software, die den Zugriff auf HPFS von DOS ermöglicht, aber dies ist nicht immer empfehlenswert.

Eine Sonderform des HPFS ist das HPFS386, das dem LAN Server beigefügt ist. HPFS386 wurde in vielen Dingen nochmals auf Performance getrimmt und erweitert die Struktur um Dateirechte, die an jedem Verzeichnis und an jeder Datei angegliedert sind. HPFS386 erfüllt alle Aufgaben, die in einer Netzwerkumgebung erforderlich sind. Mit Hilfe des HPFS 386 läßt sich sogar der Zugriff auf eine lokale Festplatte eines Mitarbeiters sperren.

HPFS386 kommt nicht beim normalen OS/2 zum Einsatz, da es in einer Einzelplatzumgebung einem »Mit Kanonen auf Spatzen schießen« gleichkommt.

3 Die Systemdateien von OS/2

von Jürgen Sting

OS/2 kennt drei wichtige Systemdateien:

– »config.sys«

– »os2.ini«

– »os2sys.ini«

OS/2 benutzt weitere Dateien, um seine Konfiguration und die Programmeinstellungen zu speichern, auch die EA-Attribute (erweiterte Attribute) werden dazu genutzt. Auf diese Dateien/Objekte werde ich in einem speziellen Abschnitt eingehen.

Während sich »os2.ini« und »os2sys.ini« mit OS/2-Bordwerkzeugen nur mit Programmierkenntnissen in RexX bearbeiten lassen, bietet sich ein Bearbeiten der »config.sys« schon mit »e.exe« oder »epm.exe« geradezu an.

»config.sys«

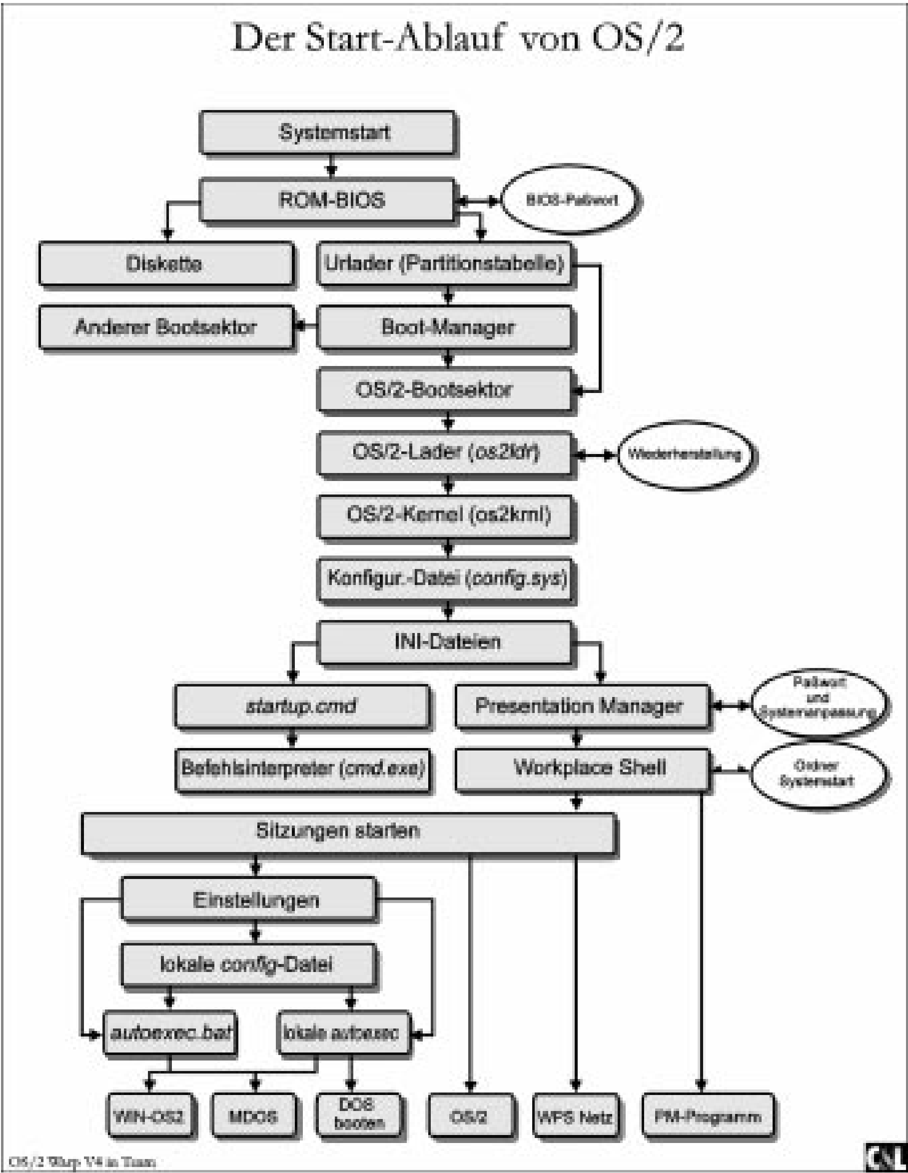
Die »config.sys« weist – verglichen mit dem DOS- und Win95-Pendant – oberflächlich einige Gemeinsamkeiten auf, in ihr werden Treiber geladen, Umgebungsvariablen festgelegt und Pfade bestimmt. Trotzdem ist es gerade für OS/2-Neulinge schwer, die »config.sys« zu verstehen.

Die Online-Dokumentation bietet zu vielen Config.sys-Einträgen gute Hinweise.

Ein sehr gutes Utility, um die OS/2-Einträge zu editieren, ist »Warp'n'go«, es handelt sich dabei um IBM Employee Written Software, wurde also von IBM-Mitarbeitern geschrieben. Änderungen in der »config.sys« werden erst mit einem Neustart des Systems aktiv!

3.1 Einträge in der Config.Sys (Basedev=)

Dies sind die wichtigsten Treiber für OS/2. Sie werden vor allen anderen Treibern geladen und sorgen für die reibungslose Kommunikation von OS/2 mit der Hardware. Diese Treiber sind aus Performancegründen oftmals sehr hardwarenah in Assembler geschrieben. Sie laufen


Der Start-Ablauf von OS/2
Systemstart
ROM-BIOS
BIOS-Paßwort
Diskette
Urlader (Partitionstabelle)
Anderer Bootsektor
Boot-Manager
OS/2-Bootsektor
OS/2-Lader (os2ldr)
Wiederherstellung
OS/2-Kernel (os2krnl)
Konfigur.-Datei (config.sys)
INI-Dateien
startup.cmd
Presentation Manager
Paßwort
und
Systemanpassung
Befehlsinterpreter (cmd.exe)
Workplace Shell
Ordner
Systemstart
Sitzungen starten
Einstellungen
lokale config-Datei
autoexec.bat
lokale autoexec
WIN-OS2
MDOS
DOS
booten
OS/2
WPS Netz
PM-Programm
OS/2 Warp V4 in Team
C&L

geschützt von jeglicher Software ab. Fehler in diesem Bereich sind sehr kritisch und können OS/2 zum Absturz bringen. Windows NT 4.0 benutzt diesen für Treiber reservierten Bereich auch für einige Teile seiner grafischen Oberfläche, Performancevorteile werden dabei mit Stabilitätseinbußen erkauft.

OS/2 akzeptiert nur die folgenden Treiberendungen, sortiert nach der Reihenfolge, in der sie geladen werden:

- .SYS
- .VSD
- .TSD
- .ADD
- .I13
- .FLT
- .DMD

OS/2 sucht **ausschließlich** in folgenden Verzeichnissen nach den Treibern:

\ (Hauptverzeichnis)

\OS2

\OS2\BOOT

3.1.1 Gebräuchliche »BASEDEV=«-Einträge

Bei fast allen Treibern können Sie den Parameter */V* setzen, der Treiber zeigt dann weitere Informationen beim Laden an. Die einzelnen Parameter finden Sie in der Online-Dokumentation bzw. auch direkt hier.

IBMKBD.SYS

Der OS/2-Tastaturtreiber

PRINT01.SYS

Der Treiber für parallelen Schnittstellen in einem PCI-oder ISA-Bus-PC (Parameter: /IRQ Schaltet den Treiber in den Interruptmodus). Der Treiber benutzt standardmäßig keinen Interrupt, sondern benutzt den sog. »polling mode«. Dies ist dann von Vorteil, wenn Sie den Interrupt z.B. für eine Soundkarte benötigen. Sollten Sie einen schnellen Drucker angeschlossen haben, verringert sich die Druckperformance, allgemein steigt die Systembelastung in diesem Modus. Dies ist bei schnellen Rechnern sicherlich zu verkraften. Allerdings sollten Sie bei langsamen Rechnern oder wenn Sie hohe Druckperformance benötigen, den Treiber mit den Interruptmodus schalten. Ein Interrupt kann nicht vorgegegeben werden.

PRINT02.SYS

Hat die gleiche Funktion wie »print01.sys«, Verwendung nur auf Rechnern, die mit dem IBM-Microchannel-Bus ausgestattet sind. Dieser Treiber kann auf Nicht-Microchannel-Rechnern gefahrlos aus der »config.sys« entfernt werden.

OS2SCSI.DMD

Die Standardschnittstellentreiber für SCSI-Treiber, die den IBM-OS/2-SCSI-Treiberspezifikationen entsprechen und nicht über »os2aspi.dmd« oder »os2dasd.dmd« angesprochen werden.

OS2DASD.DMD

Die Festplattenunterstützung (DASD = Digital Access Storage Device). Dieser Treiber muß immer geladen werden.

OS2CDROM.DMD

Die CD-ROM-Unterstützung.

OS2ASPI.DMD

Integriert das ASPI (Advanced SCSI Programming Interface) in OS/2. ASPI wurde von Adaptec entwickelt und ermöglicht es, hardwareunabhängige Treiber zu schreiben, sofern der Controller dies unterstützt. Die Controller von Adaptec unterstützen selbstverständlich ASPI, auch bei den anderen Firmen hat sich ASPI zum Standard entwickelt. SCSI hat damit ein gutes Stück an Flexibilität gewonnen.

OPTICAL.DMD

Unterstützt optische SCSI-II-Laufwerke mit Medien, die eine Sektorgröße von 512 Byte haben.

IBM1FLPY.ADD

Basistreiber für Diskettenlaufwerke. Im Normalfall (ein bis zwei Diskettenlaufwerke) brauchen Sie hier nichts zu ändern. OS/2 zeigt übrigens immer zwei Diskettenlaufwerke an.

Die Parameter von »ibm1flpy.add« sind sehr variabel und vielfältig:

Parameter	**Beschreibung**
/V	Verbose. Informationsanzeige beim Laden des Treibers
/MCA	Für Microchannel-Rechner mit ABIOS. Nicht gleichzeitig mit IBM2FLPY.ADD laden!
/A:n	Adapternummer
/IRQ:xx	Interrupt des Adapters
/P:xxx	Basisadresse des Adapters
/DMA:xx	DMA-Kanal, auf dem der Datentranfer abgewickelt werden soll
/U:	Gerätenummer

/F:xxx	Angabe der Laufwerkskapazität, wenn diese nicht korrekt erkannt wird
/CL:x	Art des Media Changed Signals. Bei jedem Laufwerkszugriff wird überprüft, ob ein Datenträgerwechsell stattgefunden hat. Der Zugriff wird dadurch langsamer. »NONE« bedeutet keine Auswertung
AT	PC-AT-Rechner-Signal wird ausgewertet
PS2	PS/2-Rechner-Signal wird ausgewertet

Ein Tip: das Entfernen dieses Treibers ist ein probates Mittel, um Zugriff auf die Diskettenlaufwerke zu unterbinden, Sie sollten den Treiber dann allerdings nicht auf der Festplatte lassen, ein findiger Anwender wird ihn einbinden, das System neu booten und hat Zugriff auf die Diskettenlaufwerke.

IBM2FLPY.ADD

Der Treiber für IBM-Microchannel-Diskettenlaufwerke. Nicht mit »ibm1flpy.add« zusammen laden. Für viele CD-ROM-Laufwerke, die nicht dem ATAPI-Standard entsprechend, liefert OS/2 Treiber mit.

XDFLOPPY.FLT

Dieser Treiber sorgt für die Erkennung von XDF-Disketten, auf denen die OS/2-Diskettenversionen ausgeliefert werden. Sofern Sie diese Unterstützung nicht benötigen, können Sie den Treiber entfernen.

SCSI-Treiber

Alle SCSI-Treiber unterstützen den Parameter */V*, d.h., sie zeigen Informationen an. Der Parameter »/A:x« bestimmt die Adapternummer, wenn mehrere Adapter installiert sind.

Tip: Sollten Sie einen CD-Wechsler installiert haben und Ihr Adapter unterstützt mehrere logische Einheiten (LUN = Logical Unit), müssen Sie den Parameter */ET* ergänzen.

DELIVERY.SYS

Sofern Sie einen IBM-SCSI-2-FAST/WIDE-Adapter einsetzen, benötigt dieser den Treiber.

AHA???X.ADD

Adaptec-SCSI-Controller-Treiber, anstelle der drei Fragezeichen steht eine die Baureihe bezeichnende Zahl.

AIC7770.ADD

Adaptec-SCSI-Controller-Treiber, für alle SCSI-Controller, die auf dem AIC7770 Chip basieren.

AIC7870.ADD

Adaptec-PCI-SCSI-Controller-Treiber, für alle SCSI-Controller, die auf dem AIC7870 Chip

basieren. Im Handel auch mit der Bezeichnung 2940 PCI-Busmaster-SCSI-Controller zu finden, ist dies die aktuellste Schöpfung aus dem Hause Adaptec, die sich auch oft als auf dem Motherboard integrierter SCSI-Controller findet.

BTSCSI.ADD

BusLogic-SCSI-Treiber.

FD*.ADD

Future-Domain-SCSI-Treiber.

IBMINT13.I13

Tip: Dieser SCSI-Treiber hilft in jedem Fall weiter, auch wenn Sie keinen entsprechenden Treiber für Ihren Adapter haben. Sie sollten in diesem Fall allerdings bei einigen Adaptern das BIOS aktivieren.

Tip: Wenn Sie nicht von einer SCSI-Platte booten, brauchen Sie das BIOS nicht beim Booten zu laden.

Tip: Booten von SCSI-Platten bei gleichzeitig installierten E-IDE-Platten ist kein Problem, wenn sie auf der ersten E-IDE-Platte den OS/2-Bootmanager installieren.

IBM2???.SYS

PCMCIA Unterstützung (PC-Card). Bis auf wenige weiter unten aufgeführte Ausnahmen nennen sich die PCMCIA-Treiber für verschiedene Notebooks so.

ICMEMCDD.SYS

Treiber für PCMCIA Flash-ROM

ICMEMMTD.SYS

Treiber für PCMCIA Flash-ROM

PCMCIA.SYS

PCMCIA Card-Services Basistreiber

RESERVE.SYS

Mit diesem Treiber können Sie bestimmte Systemressourcen reservieren, nähere Informationen entnehmen Sie bitte der OS/2-Online-Dokumentation.

IDE-Treiber

IDE ist heute, trotz bekannter Einschränkungen, immer noch die am weitesten verbreitete Schnitstelle für Festplatten-Systeme. An dieser Stelle lassen sich daher nicht nur die meisten Optimierungen vornehmen, sondern es treten dort auch leicht Fehler auf. Die Installation eines CD-ROM Laufwerks, einer weiteren Festplatte, gar eines zweiten IDE-Adapters oder mehrerer CD-ROM-Laufwerke macht es notwendig, selbst Hand an den folgenden Treibern anzulegen. Zwar ist die Hardwareerkennung in zwischen sehr ausgereift, aber Sie sollten wissen, was zu tun ist, wenn Warp 4 es nicht weiß.

IBM1S506.ADD

Festplatten-Basistreiber für alle MFM-, RLL-, ESDI-, IDE-, E-IDE- und ATAPI-Festplatten/Laufwerke.

Parameter	Beschreibung
/V	Verbose Mode. Zeigt Statusmeldungen und Informationen zu den Laufwerken.
/A:n	Adapternummer, normalerweise immer 0, wenn sie 2 Adapter installiert haben, 1.
/I	Adapter ignorieren
/P:xxx	Portadresse
/IRQ:nn	Interrupt
/U:n	Gerätenummer 0 Erstes Gerät am Adapter 1 Zweites am Adapter angeschlossenes Gerät
/T:n	Maximale Wartezeit, wenn ein Fehler aufgetreten ist.
/GEO:(C,H,S,W)	Laufwerksgeometrie. Den korrekten Wert finden Sie in der Dokumentation Ihrer Festplatte.
C	Cylinder. Zylinderanzahl der Festplatte
H	Heads. Anzahl der Schreib-/Leseköpfe der Festplatte
S	Sectors. Sektorenzahl
W	Write Precompensation. Wird nur von sehr alten Festplatten benötigt, moderne Festplatten ignorieren diesen Wert.
/!SMS	Disable Multiple Sektor Support. Durch den heute in modernen Festplatten enthaltenen Cache kann die Platte mehrere Sektoren auf einmal einlesen. Bei älteren Platten deaktivieren gegebenenfalls diese Unterstützung.
/!DM	Disable DASD-Support (DASD= Digital Access Storage Device). Der DASD Manger ignoriert dieses Gerät.
/LBA	Logical Block Access. Neuere BIOS-Versionen unterstützen den LBA-Modus, in dem die 512-Mbyte-Grenze bei E-IDE-Festplatten aufgehoben wird, E-IDE-Festplatten können bis zu 7 Mbyte groß sein und werden von DOS korrekkt erkannt.

Tip: Bei älteren BIOS-Versionen müssen die bootfähigen Partitionen unterhalb von 512 Mbyte liegen, darüber hinaus kann kein FAT-Dateisystem benutzt werden, hier braucht es das professionelle HPFS.

Des FAT-Dateisystem kann die 512-Mbyte-Grenze überwinden, durch Treiber und spezielle »Kniffe« ist dies möglich. Grundsätzlich ist davon abzuraten, FAT Partitionen größer als 128 Mbyte anzulegen. Die Clustergrößen und die damit einhergehende Verschwendung von Plattenplatz ist umso größer, je größer Sie die Partition wählen. HPFS nutzt immer optimale 512-Byte-Cluster (Sektor=Cluster). Nutzen Sie, wo immer möglich HPFS. Auch das mit neuen DOS/Win- Versionen mitgelieferte VFAT ist keine Lösung, es hat alles Schlechte von FAT

geerbt. HPFS ist auf jeden Fall die professionelle Lösung. Die Performancevorteile auf großen Platten sind auch nicht zu unterschätzen.

Tip: Die ersten CD-ROM-Laufwerke nach dem ATAPI-Standard waren nicht voll kompatibel zu diesem. Sollten Sie Probleme mit einem solchen Laufwerk haben, halten Sie in einer Mailbox Ausschau nach folgenden Archiven: ATAPI.* TEMPFIX.*, MITFX.*.

Tip: Bei zwei CD-ROM-Laufwerken im Rechner setzen Sie jeweils erst den IBM1S506-Treiber und dann den entsprechenden CD-ROM-Treiber, wobei Sie dem IBM1S506-Treiber die Daten wie Adapternummer und Gerätenummer als Parameter übergeben.

**.FLT-Treiber*

Diese Treiber sorgen z.B. bei CD-ROM-Laufwerken für den korrekten Zugriff und sind unbedingt notwendig.

3.2 Installierbare Dateisysteme

Die praktischsten Features an OS/2 sind sicherlich die installierbaren Dateisysteme. Das FAT-Dateisystem ist in den Kernel integriert und zählt daher nicht zu den installierbaren Dateisystemen. Es wird direkt beim Start benötigt, denn zur Installation benötigen Sie immer noch Disketten, die mit dem speziell für Sie entwickelten Dateisystem formatiert sind – FAT.

Das bekannteste installierbare Dateissystem ist HPFS, direkt danach kommt CDFS (für CD-ROM). Inzwischen ist die Zahl der installierbaren Dateissysteme für OS/2 stark angewachsen. Komprimierende Dateisysteme sind ZIPStream, verschlüsselnde sind z.B. Cryptstream oder UNIX-Dateisysteme. Interessant sind auch virtuelle Dateisysteme, wie das TVFS (Toronto Virtual File System). Sie ermöglichen es, bestimmte Verzeichnisse zu einer Festplatte zu machen oder bestimmte Platten zu einem Verzeichnis zu konfigurieren. Wenn Sie häufig Laufwerke wechseln (ZIP-Laufwerke), kann dies zu einem unentbehrlichen Helfer werden.

IFS=HPFS.SYS

Der Treiber für das installierbare Dateisystem »High Performance File System«. Hier können Sie eine Menge optimieren, nicht nur die Cache-Größe wird hier eingestellt, sondern auch andere Performance-beinflußende Parameter stehen Ihnen zur Verfügung.

/C:xxxx

Cache. Mit diesem Parameter können Sie die Cache-Größe einstellen. Werte bis 2 Mbyte bringen spürbare Steigerungen. Der Wert wird in Kbyte angegeben. Bei sehr knappem Speicher sollten Sie ihn auf 128 oder gar 64 Kbyte reduzieren.

OS/2 kennt kein Smartdrive, ein goßer Cache steigert die Performance nicht, sondern bremst ein System gnadenlos in die Knie. Stellen Sie nie mehr als 10 Prozent Ihres Hauptspeichers insgesamt als Cache zur Verfügung. Bei 32 Mbyte sollten Sie nicht mehr als 3,2 Mbyte für Cache opfern. Mehr als 3-4 Mbyte sind nicht ratsam, es sei denn, Sie setzen Spezialanwendun-

gen ein, die dadurch beschleunigt werden. Der Cache für HPFS-/FAT-/CD-ROM-Laufwerke läßt sich getrennt einstellen.

/AUTOCHECK:

Dieser Parameter sorgt für die Überprüfung Ihrer Festplatten, standardmäßig tauchen hier alle bei der Installation vorhandenen Laufwerke auf, ergänzen Sie evtl. weitere HPFS-Laufwerke. OS/2 setzt beim Booten automatisch ein sog. »Dirty Flag«, dies wird zurückgesetzt, wenn ein Systemabschluß korrekt durchgeführt wird oder das System per Warmstart ([Strg]+[Alt]+[Entf]) neu gestartet wird, OS/2 schließt dann alle Dateien korrekt. Bei Stromausfall schalten Sie den Rechner einfach aus. Bei einem sonstigen Defekt, der OS/2 zum Absturz bringt, wird OS/2 beim nächsten Booten die Festplatte einem eingehenden Test unterziehen.

Tip: Ein »+« vor dem Laufwerk sorgt dafür, das es immer geprüft wird. Empfehlenswert, wenn Sie eine Platte mit wichtigen Daten haben und immer sicher sein wollen, daß das Dateisystem in korrektem Zustand ist.

Es gibt noch einen undokumentierten Parameter: ein Sternchen »*« sorgt dafür, daß grundsätzlich alle Laufwerke, wenn nötig, überprüft werden. Ein weiterer undokumentierter Parameter sorgt dafür, daß OS/2 keinesfalls die Platte prüft, dies kann schlimme Schäden bis hin zu Datenverlusten zur Folge haben, ich möchte diesen hier deshalb nicht erwähnen.

Beispiele:

```
/AUTOCHECK:CD+E
```

Die HPFS-Laufwerke CD werden, wenn nötig überprüft. E wird immer überprüft.

```
/AUTOCHECK:*
```

OS/2 prüft gegebenenfalls alle HPFS-Laufwerke.

/CRCL:xx

Gibt die Größe der Cacheline in Kbyte an. Werte zwischen 2 und 64 Kbyte, die durch 2 teilbar sind, sind zulässig. Moderne Festplatten besitzen einen Cache, daher können Daten im voraus eingelesen werden, praktisch auf Verdacht, da es wahrscheinlich ist, daß die nächsten Daten innerhalb der gelesenen Datei als nächstes benötigt werden. Diesen Parameter können Sie auf den Maximalwert setzen, die Performance-Steigerung ist spürbar.

/F:

Identisch mit /F: bei CHKDSK, er legt das CHKDSK-Level fest. Standard ist 2.

Tip: Sofern Sie Ihre HPFS-Festplatte formatiert haben, kann CHKDSK /F:3 die formatierten Daten wiederherstellen, in vielen Fällen ohne Verluste. Sie sollten daher Ihre Festplatte entweder nur mit CHKDSK /F:2 prüfen, oder eine Formatierung mit FORMAT d: /L durchführen, sonst kann es zu dem Effekt kommen, daß Daten unerwünschterweise wiederhergestellt werden.

Rückgabewert	Fehler
0	keine Korrektur, nur Erkennung
1	Fehlererkennung und Behebung von Dateisystemfehlern
2	Fehlererkennung, Korrektur von Dateisystemfehlern und Wiederherstellung defekter Strukturen im momentan benutzen Plattenbereich
3	Fehlererkennung, Korrektur von Dateisystemfehlern und Wiederherstellung defekter Strukturen auch im unbenutzen Plattenbereich

IFS=CDFS.SYS

Ohne diesen Treiber kann OS/2 auf CD-ROM-Laufwerke zugreifen, »versteht« deren Inhalt jedoch nicht.

Der CDFS-Treiber kennt ähnliche Parameter wie der HPFS-Treiber:

Parameter	Name	Beschreibung
/C:Wert	Cache	Cache-Größe in 64-Kbyte-Blöcken, der Standardwert von 2 besagt also 128 Kbyte Cache.
/M:Wert	–	Größe der Cacheline, Sektoren der aktuellen Datei, die auf Verdacht schon im voraus geladen werden.
/Q	Quiet	Keine Meldung bei Systemstart.
/P:Wert	–	Keine Bedeutung für den Normalanwender
/K	–	Keine Bedeutung für den Normalanwender

3.3 Die SET-Einstellungen (SET=)

```
SET AUTOSTART=PROGRAMS,TASKLIST,FOLDERS,CONNECTIONS,WARPCENTER,LAUNCHPAD
```

Die aufgeführten Komponenten werden von OS/2 oder besser der WPS bei einem Systemstart wieder geladen.

Hier die Parameter und deren Beschreibung:

PROGRAMS

Alle beim Systemabschluß geladenen Programmobjekte werden gestartet. Achtung: Programme im Systemstartordner werden trotzdem gestartet. Tip: Drücken Sie beim Erscheinen des ersten grafischen Bildschirms, also wenn der schwarze Hintergrund verschwindet,

[STRG]+[SHIFT]+[F1]. So wird unterbunden, daß die Programme aus dem Systemstartordner gestartet werden.

TASKLIST

Laden der Fensterliste. Achtung: die Fensterliste sollten Sie nie entfernen.

FOLDERS

Die geöffneten Ordner werden wieder geöffnet. Achtung: Die Arbeitsoberfläche ist nichts anderes als ein Ordner!

CONNECTIONS

Alle Netzwerkverbindungen, die beim letzten Systemabschluß aktiv waren, werden, sofern möglich, wiederhergestellt.

WARPCENTER

Der Ersatz für die bekannte Klickstartleiste, der der Windows95-Startleiste ähnlich sieht, sie jedoch an Funktion übertrifft, kann mit diesem Schlüsselwort gestartet werden. Gefällt Ihnen die Startleiste nicht, entfernen Sie dieses Schlüsselwort. Sie können sie dann immer noch im Systemordner aufrufen.

LAUNCHPAD

Die Klickstartleiste aus Warp 3 existiert noch und kann so wieder normal genutzt werden.

SET BOOKSHELF=

In diesen Verzeichnissen sucht »view.exe« nach den OS/2-INF-Dateien (Online-Dokumentation). Legen Sie sich ruhig ein Verzeichnis an, in dem Sie alle Inf-Dateien »aufbewahren«, die im sich Laufe der Zeit ansammeln. Viele interessante Informationen rund um OS/2 tauchen inzwischen in diesem Format auf. Es bietet den Vorteil, daß jeder OS/2-Anwender die damit erstellten Dateien nutzen kann, Grafiken und formatierte Texte sind damit möglich und dem ganzen kann eine übersichtliche Stuktur gegeben werden.

Beispiel:

```
SET BOOKSHELF=D:\OS2\BOOK;D:\MMOS2;D:\SYSTEM\BOOK
```

Nach der Installation sieht die Zeile meistens so aus.

```
SET COMSPEC=D:\OS2\CMD.EXE
```

Hier geben Sie den Befehlszeileninterpreter für OS/2 an, besser gesagt, hier steht »cmd.exe«, das OS/2-Pendant zum DOS-Commamnd.com. Sie können hier auch ein Alternativprogramm wie z.B. 4OS2 einsetzen (z.B. »d:\4os2\4os2.exe«).

```
SET DELDIR=Laufwerk:\Pfad,Größe; Laufwerk:\Pfad,Größe ...
```

Standardmäßig werden Sie diese Zeile als »REM SET DELDIR=...« finden, es handelt sich dabei um die »Undelete«-Funktion von OS/2. OS/2 verschiebt die Dateien in das angegebene Verzeichnis, sobald die vorgegebene Größe erreicht wird; löscht OS/2 die Dateien, die am sich

längsten im Verzeichnis befinden, geschieht dies nach dem Prinzip FIFO (First In, First Out). Dies verlangsamt die Löschoperationen nicht unerheblich, ermöglicht Ihnen jedoch mit Undelete.Exe eine schnelle, sichere Wiederherstellung. Sie können das Verzeichnis beliebig benennen.

Beispiel:

```
SET DELDIR=C:\Ablage_P,5120;D:\Papierkorb,20480;E:\Papierkorb,1024;F:\Papier-
korb,4096; ...
```

SET DIRCMD=

Identisch mit seinem DOS-Pendant, können Sie hier vorgeben, welche Parameter der DIR-Befehl immer benutzen soll.

SET DPATH=

Verlängert man DPATH, kommt man zu DATAPATH, die Übersetzung »Datenpfad« trifft den Sinn recht genau. In den in DPATH angegbenen Verzeichnissen sucht OS/2 nach den Datendateien zu entsprechenden Programmobjekten.

SET DSPPATH=

Hier findet OS/2 die DSP-Dateien für Ihre Soundkarte. DSP steht für »Digital Sound Processing«, die Marketingabteilung mancher Hersteller verkauft DSP auch unter dem Label ASP oder Advanced Signal Processing.

SET EPMPATH=

Sie ahnen es schon, der erweiterte Editor findet sich in diesem Pfad.

SET GLOSSARY=

Pfad für das OS/2-Glossar.

SET HELP=

OS/2 sucht in diesem Pfad nach den *.HLP-Dateien (Hilfedateien). Die Hilfedateien sind nicht zu verwechseln mit der Onlinedokumentation. Die Dateistruktur HLP/INF ist bis auf wenige Byte identisch, doch werden Hilfedateien mit Programmen ausgeliefert und können aus diesen heraus z.B. mit [F1] aktiviert werden. View.Exe zeigt Hilfedateien nicht an. Inzwischen gibt es viele Programme, die HLP-Dateien anzeigen, eins liegt z.B. dem Warp Server bei. »Viewhlp«, auch FM/2, ein Shareware-Dateimanager für OS/2, öffnet Hilfedateien auf Doppelklick. Sie können alle Hilfedateien in einem Verzeichnisobjekt konzentrieren, tragen Sie den Pfad dann in die SET HELP-Variable ein.

SET IPF_KEYS=

Diese Variable wird von der »Information Presentation Facility« (IPF) alias »view.exe« ausgewertet.

OS/2 unterstützt »SBCS« (Single Byte Character Set) und »DBCS« (Double Byte Character Set). Die deutsche Sprache ebenso wie fast alle europäischen Sprachen nutzen die lateinische

Schrift, von Umlauten einmal abgesehen, ist die Vielfalt der Schriftzeichen nicht allzu groß. Die mit einem Byte darstellbaren 256 Schriftzeichen reichen für die Darstellung. Im asiatischen Raum jedoch sind tausende Schriftzeichen keine Seltenheit, für diese Zwecke wurde DBCS entwickelt, mit 2 Byte lassen sich 65536 Schriftzeichen darstellen.

SET KEYS=

Schaltet die History-Funktion der Befehlszeile ein (ON) oder aus (OFF). Sie können einfach mit den Pfeiltasten oben/unten durch bereits benutzte Befehle »blättern«. Erweiterte Editierfunktionen werden dadurch ebenfalls aktiviert, d.h., sie können mit den Pfeiltasten links/rechts durch die Eingabezeile »gehen«.

SET MMBASE=

Verzeichnis für die OS/2 Multimediaerweiterungen. OS/2 Warp 4.0 installiert übrigens erstmals die Multimedia-Erweiterungen standardmäßig. Dies macht auch Sinn, da heutzutage kaum noch ein Nutzer auf die Multimedia-Erweiterungen verzichtet und sie für bestimmte Anwendungen Vorraussetzung sind, z.B. die Spracherkennung oder JAVA.

SET MENUSTYLE=LONG

Dieser Eintrag taucht in der »config.sys« normalerweise nicht auf. Wenn Sie kurze Menüs wünschen (ohne löschen, anordnen...), benutzen Sie den Schalter »SHORT«.

SET NCDEBUG=4000

Einige sehr alte Lotusanwendungen nutzen diesen Parameter für Ihre Makrofunktionen. Sollten Sie keine derartigen Anwendungen/Makros nutzen, können Sie die Zeile löschen.

SET OS2_SHELL=D:\OS2\CMD.EXE BATCH.CMD

Der Standardbefehlszeileninterpreter, kann z.B. auch 4DOS sein. Wenn Sie ein ReXX-Programm mitangeben, wird dieses bei jedem Aufruf von »cmd.exe« ausgeführt.

SET PATH=

Programmpfad für OS/2- und WinDOS-Anwendungen. Alle Programmobjekte, die nicht im aktuellen Verzeichnis gefunden werden, werden in den Verzeichnissen in der Pfadangabe gesucht. Bitte beachten Sie, daß OS/2 »der Reihe nach« vorgeht, eine weiter vorne im Pfad stehende Angabe wird also vor der zum Schluß stehenden bearbeitet.

SET RESTARTOBJECTS=

Gibt an, welche Objekte beim letzen Systemabschluß aktiv waren und unter welchen Bedingungen OS/2 sie wieder aktiviert.

Parameter:

YES	Alle Objekte werden gestartet.
NO	Es werden keine Objekte gestartet.
STARTUPFOLDERSONLY	Objekte im Startupfolder werden gestartet.

REBOOTONLY

Die Objekte werden nicht bei einem Neustart der WPS neu gestartet, sondern nur nach einem Reboot. REBOOTONLY kann mit den anderen Anweisungen kombiniert werden.

SET RUNWORKPLACE=D:\OS2\PMSHELL.EXE

Die Standard-Benutzeroberfläche. Wenn Sie auf die WPS verzichten wollen, können Sie z.B. SET RUNWORKPLACE=D:\OS2\CMD.EXE in Ihre »config.sys« einsetzen. Der Presentation Manager (grafische Anwendungen) steht Ihnen weiterhin zur Verfügung.

SET SOMDDIR=

Verzeichnis für die (D)SOM-Erweiterung. DSOM steht für »Distributed System Object Model«.

SET SOMDIR=

Verzeichnis für die DSOM-Runtime-Module

SET SYSTEM_INI= und SET USER_INI=

Diese beiden Einträge definieren die beiden wichtigen INI-Dateien von OS/2.

SET VIDEO_DEVICES=

Wird für die Grafikkonfiguration benötigt.

SET VIO_SVGA=

Angabe der Basis-Videotreiber, die das System nutzt.

3.4 Sonstige OS/2-Einstellungen

AUTOFAIL=NO

Dieser Parameter unterdrückt, sofern man YES einsetzt, die Anzeige von Fehlermeldungen, es wird immer Ja/OK angenommen.

BUFFERS=30

Anzahl der 512 Byte großen Festplattenpuffer für Zugriffe auf die Festplatte <= 512 Byte. Werte zwischen 1 und 100 sind zulässig. Ein höherer Wert kann die Systemleistung insbesondere bei der Arbeit mit kleinen Dateien oder dem Dateisystem FAT erhöhen. Auf Systemen mit wenig Speicher sollte man »Buffers« reduzieren z.B. auf 10.

CODEPAGE=437,850

Für die verschiedenen nationalen Sprachen sind unterschiedliche Zeichentabellen erforderlich. Standard für Deutschland ist 437 (deutsch, national). Die sekundäre Zeichentabelle, auf die

Sie mit CHCP wechseln können, ist 850 (deutsch, international).

DISKCACHE=xxxx,LW,yy,AC:d

Cache für nicht wechselbare FAT-Laufwerke (also z.B. keine Disketten).

xxxx Größe des Cache, 48 Kbyte bis 14400 Kbyte

oder:

D OS/2 bestimmt die Größe des Cachespeichers bei jedem Systemstart neu:

Empfohlene Werte für das Caching:

1 – 4 Mbyte RAM	48 Kbyte
4 – 5 Mbyte RAM	48 Kbyte bzw. 64 Kbyte
5 – 6 Mbyte RAM	128 Kbyte
6 – 8 Mbyte RAM	512 Kbyte
> 8 Mbyte RAM	10% des Hauptspeichers, max. 4 Mbyte

yy Anzahl der Sektoren, die maximal bei einem Lesezugriff gelesen werden sollen. Werte zwischen 4 (Standard) und 128 sind zulässig. Als praktikabel erweist sich ein Wert von 32.

LW (Lazywrite) Aktivierung der Schreibverzögerung

AC:C+D (Autocheck). OS/2 setzt auch auf FAT-Partitionen ein dirty-Flag (siehe HPFS.IFS). Sollte das System nicht korrekt heruntergefahren worden sein, so überprüft OS/2 automatisch die eingetragenen Festplatten. Die OS/2-Startpartition, sofern FAT-formatiert, taucht hier immer auf. Durch ein Plus vor einem Laufwerk wird dieses immer geprüft.

IOPL= (Input Output Privilege Level)

Parameter:

YES Allen Programmen wird direkter Zugriff auf die Hardware erlaubt.

NO Direkter Zugriff auf die Hardware nicht erlaubt, Zugriff nur über Treiber.

LASTDRIVE=

Dieser Befehl gibt das letzte Laufwerk an, das sie verwenden wollen. Weitere Laufwerke können nicht verwendet werden.

LIBPATH=

Hier sucht OS/2 nach den *.DLL-Dateien. Häufig weigert sich ein Programm zu starten, wenn die entsprechenden DLL-Dateien nicht vorhanden sind, bzw. sich nicht im Libpath befinden. Viele Sharewareprogramme benötigen die »vrobj.dll«. Sie sollten diese nur einmal in der neuesten Version auf der Platte haben, z.B. im os2\dll-Verzeichnis.

MAXWAIT=

OS/2 teilt einem Programm nach entsprechender Wartezeit in Sekunden eine höherer Priorität zu. Anwendungen, die auf eine Eingabe/Ereignis warten, bekommen keine höhere Priorität zugeteilt. Bei schnellen Rechnern können Sie diesen Wert auf 1 setzen, das Multitasking wird dadurch flüssiger.

MEMMAN=

Arbeitsweise des *MEM*ory*MAN*agers

SWAP

Verwendung von virtuellen Speicher

NOSWAP

Keine Verwendung von virtuellem Speicher. Bitte diesen Parameter nicht verwenden, er kann zu Fehlern führen, wenn der Speicher nicht ausreicht.

MOVE/NOMOVE

Befehl wird nicht mehr verwendet.

COMMIT

Für die Kompatibilität zu OS/2-1.X-Programmen, die Speicher, den Sie anfordern, auch komplett zur Verfügung gestellt haben wollen. Anwendungen für OS/2 2.X und höher fordern zwar auch Speicher an, dieser wird aber nach Bedarf zur Verfügung gestellt.

PROTECT

Verwendung der erweiterten Speicherschutzmechanismen von OS/2

NOPROTECT

Keine Verwendung der erweiterten Speicherschutzmechanismen von OS/2. Im Normalbetrieb nicht verwenden.

PAUSEONERROR=YES

YES	Tritt ein Fehler bei der Bearbeitung der »config.sys« auf, stoppt OS/2 hier, durch Bestätigen der Fehlermeldung können Sie den Bootvorgang fortsetzen.
NO	OS72 gibt bei einem Fehler eine Fehlermeldung aus, der Bootvorgang wird fortgesetzt. Bei unbeaufsichtigten Systemen kann dieser Parameter sinnvoll sein, z.B. bei Mailboxen, die nachts einen Reboot durchführen.

PRINTMONBUFSIZE=134,134,134

Puffergröße in Byte für die parallelen Schnittstellen LPT1, LPT2 und LPT3. Werte zwischen 134 und 2048 Byte sind zulässig. Standardwert ist gleichzeitig das Minimum von 134 Byte.

PRIORITY=DYNAMIC

Prioritätensteuerung von Threads

DYNAMIC

OS/2 legt die Priorität selbst fest, dadurch wird ein stabiles und flüssiges Multitasking möglich. Da kein Programm den Prozessor für sich alleine beanspruchen kann, werden selbst Programme mit geringer Priorität ausgeführt. Diese Einstellung sollte nicht verändert werden.

ABSOLUTE

OS/2 greift nicht in die Prioriätssteuerung ein, die Programme werden nach Priorität ausgeführt. Es kann passieren, daß Programme mit geringerer Priorität nicht mehr ausgeführt werden. Insgesamt bringen Sie OS/2-Multitasking damit fast auf Windows-Niveau herunter.

PRIORITY_DISK_IO=YES

Die im Vordergrund laufende Anwendung erhält Priorität beim Zugriff auf die Massenspeicher. Das verkürzt die Antwortzeiten.

Bei NO erhalten die Vorder- und Hintergrundanwendungen die gleiche Priorität beim Zugriff.

PROTSHELL=D:\OS2\PMSHELL.EXE

Protectedmode-Shell. Sofern Sie kein Ersatzprogramm einsetzen, lassen Sie diesen Eintrag unverändert.

REIPL=

Mit »ON« wird nach einem TRAP das System neu gestartet, bei »OFF« stoppt der TRAP das System, eine detaillierte Fehlermeldung gibt Auskunft, wo der Fehler aufgetreten ist (Standard).

SWAPPATH=x:\ 20480 40960

Pfad, Warngröße und Anfangsgröße der »swapper.dat«. Der erste Wert ist die sog. Warngröße. Wenn weniger als dieser Wert auf der Partition der Datei »swapper.dat« frei ist, warnt OS/2 mit einem entsprechenden Hinweis. Beenden Sie dann einige Anwendungen, brechen Sie einen Vorgang ab oder arbeiten Sie einige Minuten nicht mit dem System, schrumpft die »swapper.dat« wieder. Sollte dies keine Wirkung zeigen, speichern Sie die Dateien ab und booten Sie das System neu. Der zweite Wert, die sogenannte Anfangsgröße, bestimmt die Startgröße der»swapper.dat« beim Booten. Sie sollten den Wert so hoch wählen, wie normalerweise Ihre Speicheranforderungen sind. Dies erhöht die Performance nicht unerheblich.

SUPPRESSPOPUPS=d

Die Fehlermeldungen werden unterdrückt und stattdessen in die Datei »popuplog.os2« auf Laufwerk D protokolliert.

THREADS=256

Anzahl der Threads, die das System maximal verwalten soll. Werte zwischen 32 und 4096 sind zulässig, reduzieren Sie den Wert bei knappen Speicher auf z.B. 64.

TIMESLICE=x,y

Einstellung der Zeitscheibe. Sie sollten die vorgegebenen Werte nicht verändern.

x	Für x Millisekunden erhält ein Prozeß Rechenzeit, bevor ein Prozeß von gleicher Priorität Rechenzeit zugeteilt bekommt. Der Standardwert ist 32. Wertebereich: 32 bis (y-1)
y	Spätestens nach y Millisekunden erhält ein anderer Prozeß Rechenzeit. Hier muß der Wert von y logischwerweise mindestens X+1 sein, maximal jedoch 65536.

3.5 Sonstige DOS-Einstellungen

Identisch mit den DOS-Einstellungen unter Standard-DOS.

BREAK=

Ist der Eintrag »OFF« kann ein Programm nur mit [Strg]+[C] unterbrochen werden, wenn es auf eine Eingabe wartet. Bei Eintragung von »ON« unterbricht das Programm sofort.

DOS= OS/2 kann DOS in den oberen Speicherbereich auslagern und Speicherbereich über 640 Kbyte zur Verfügung stellen, wenn Anwendungen dies fordern. Lassen Sie ansonsten die Standardeinstellungen unverändert. Sollte ein Programm mehr Speicher benötigen, können Sie in den Einstellungen des Programmobjekts sowohl DOS in den oberen Speicher laden als auch Speicher über 640 Kbyte zur Verfügung stellen.

Parameter:

LOW	DOS wird in die ersten 640 Kbyte des Arbeitsspeichers geladen, dort belegt es zirka 640 Kbyte.
HIGH	DOS wird in die HMA (High Memory Area) geladen, dadurch werden die 640 Kbyte »normaler« DOS-Speicher entlastet.
NOUMB	OS/2 stellt keine UMBs zur Verfügung, d.h., es steht kein Speicher über 640 Kbyte zur Verfügung.
UMB	OS/2 stellt UMBs zur Verfügung, sie können den Speicher über 640 Kbyte nutzen.

FCBS=x,y

Alte DOS-Progamme nutzen File Control Blocks (FCBs), Sie können diesen Wert bequem auf 2,1 reduzieren.

FILES=xx

Anzahl der Dateien, die ein DOS-Programm gleichzeitig öffnen darf.

RMSIZE=640

Der Basisspeicher, den OS/2 der DOS-Emulation zur Verfügung stellt.

SHELL=D:\OS2\MDOS\COMMAND.COM D:\OS2\MDOS /P

Der DOS-Kommandointerpreter, Sie können hier auch alternative Kommandointerpreter wie NDOS oder 4DOS verwenden.

3.6 Sonstige Aufrufe

CALL

Ein Programm wird ausgeführt, nach der Ausführung wird die »config.sys« weiterbearbeitet.

RUN

Ein Programm wird im Hintergrund ausgeführt, die »config.sys« wird weiter bearbeitet.

3.7 Tips um die Systemdateien

Beschädigung und Totalverlust

»os2.ini« und »os2sys.ini« sind die wichtigsten OS/2-Konfigurationsdateien nach der nur beim Booten benötigten »config.sys«. Hier können Änderungen schon während des Betriebs ohne Neustart wirksam werden.

Diese Dateien sollten Sie im Normalfall nicht bearbeiten, es gibt jedoch ein paar Tips zur »Wartung« dieser Dateien. Sie sollten sich dazu einen INI-Editor zulegen. Für normale Zwecke reicht der in FM/2 integrierte, für gehobene Ansprüche ist »IniMaint« empfehlenswert.

Bestimmte Änderungen können Sie nur vornehmen, wenn Sie Ihr System von den Dienstprogrammdisketten starten.

Die »os2.ini« und die »os2sys.ini« werden mit Hilfe der Dateien »ini.rc« bzw. »inisys.rc« erzeugt. Sollten die INI-Dateien beschädigt werden (zum Beispiel durch Stromausfall), sollten Sie mit

```
MAKEINI OS2.INI INI.RC
```

bzw.

```
MAKEINI OS2SYS.INI INISYS.RC
```

die Initialisierungsdateien wieder reparieren, genauer gesagt wieder erzeugen. Die RC-Dateien sind übrigens reines ReXX. Wenn Sie sich mit ReXX auskennen, können Sie einfach Ihre eigenen *.RC-Dateien erzeugen.

Passwort vergessen

Sollten Sie Ihr Passwort für die WPS vergessen haben, dann können Sie mit

```
MAKEINI OS2.INI LOCK.RC
```

dieses wieder zurücksetzen. Brauchen Sie einen wirklich sicheren Weg, Ihren PC vor unbefugter Benutzung zu schützen, verwenden Sie ein Programm, das auf die in OS/2 eingebauten Sicherheits-APIs aufsetzt. Hier wird es mit Sicherheit auch Sharewareprodukte geben.

Pflege

Sie sollten regelmäßig die Einträge von Programmen in den Ini-Dateien untersuchen. Viele Programme installieren Sie nur einmal, um sie zu testen, gute Programme entfernen sich und auch den Eintrag aus den INI-Dateien selbst. Der häufig verwendete IBM-Installer macht dies standardmäßig. In allen anderen Fällen sammelt sich in den Inis schnell ein Haufen »Müll«. Löschen Sie diese Einträge, wenn Sie sie nicht mehr eindeutig einem Programm zuordnen können.

Sicherungskopie

Die INI-Dateien werden beim Archivieren automatisch gesichert, Sie können aber auch Kopien davon einfach auf eine Diskette ablegen.

Editieren

Die entsprechenden guten Editoren verfügen über eine gute Dokumentation der Ini-Einträge, doch im Normalfall sollten Sie nichts ändern. Nicht nur mit Editoren, auch mit ReXX können Sie Änderungen an den Inis vornehmen, entnehmen Sie auch hier der Online-Dokumentation die Vorgehensweise.

4 Die Arbeitsoberfläche

von Dimitrios Tsoukalas

4.1 Einführung in die objektorientierte Welt

Dieses Kapitel beschreibt die grafische Benutzerumgebung von OS/2 Warp Version 4, die Arbeitsoberfläche (auch WPS, vom englischen »Workplace Shell«), und deren Bedienung. Auf der OS/2-Arbeitsoberfläche, die einem Büroarbeitsplatz nachempfunden wurde, werden Sie sich sicher schnell zurechtfinden. In diesem Kapitel gehen wir nur auf die Bedienung mit Tastatur und Maus ein. Zur Spracherkennung (VoiceType) lesen Sie bitte das zugehörige Kapitel in diesem Buch.

4.1.1 Objektorientiertes Arbeiten

Eine der großen Stärken von OS/2 ist die Objektorientiertheit (OO). Ein Objekt ist eigentlich alles, was Sie auf der Arbeitsoberfläche sehen oder auf die Arbeitsoberfläche holen können. Eine Datei, in der Sie einen Brief an Ihren Steuerberater abgespeichert haben, ist genauso ein Objekt wie beispielsweise das Programm, mit dem Sie diesen Brief geschrieben haben – oder in einem Dateifenster auch das Diskettenlaufwerk.

Jedes Objekt kann durch bestimmte Aktionen, die sogenannten »Methoden«, bearbeitet werden. Die möglichen Methoden hängen vom Objekt ab. So werden Sie das Objekt »Brief an den Steuerberater« mit einer Textverarbeitung ändern, über den Drucker ausgeben oder löschen wollen, während Sie eine Diskette zum Beispiel formatieren können.

Was bringt das objektorientierte Konzept nun für den Anwender? Das OO-Konzept unterscheidet sich vom programmorientierten (PO) Konzept zunächst in der Reihenfolge des Ablaufs.

Programmorientiert heißt: Sie starten erst das Programm und wählen dann die zu bearbeitenden Daten aus.

Objektorientiert heißt: Sie wählen erst die Daten und dann aus den möglichen Methoden die richtige Bearbeitung.

Dieser Unterschied mag zunächst unwichtig erscheinen, durch ihn wird aber erst die komfortable Arbeitsweise der Arbeitsoberfläche möglich. Zudem ist der objektorientierte Weg näher am wirklichen Leben. Ein Beispiel mag das verdeutlichen: Sie wollen einen Nagel in die Wand einschlagen, um ein Bild aufzuhängen.

OO: Sie nehmen das Objekt (Nagel) und suchen sich ein passendes Werkzeug (meist ein Hammer).

PO: Sie nehmen das Werkzeug (Hammer) und suchen sich ein passendes Objekt zum Verarbeiten (beispielsweise einen Nagel).

4.2 Die Basisobjektklassen

Als Anwender werden Sie am häufigsten den folgenden vier Objektklassen begegnen:

4.2.1 Das Datendateiobjekt



Abb. 4.1: Datendateiobjekt

Es handelt sich hierbei um ein Objekt, das Daten (Texte, Bilder, Musik, usw.) auf einer physikalischen Einheit (Festplatte, Diskette, ...) repräsentiert bzw. darstellt.

4.2.2 Das Ordnerobjekt

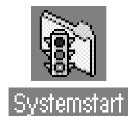


Abb. 4.2: Das Ordnerobjekt

Die Funktion des Ordners erklärt sich durch den Namen fast von selbst. Ein Ordner ist ein Behälter, der andere Objekte enthalten kann, auch andere Ordner. So werden Ordner normalerweise verwendet, um Datendateien aufzubewahren. Im Dateisystem entspricht ein Ordner einem Verzeichnis.

4.2.3 Das Programmobjekt



Abb. 4.3: Das Programmobjekt

Das Programmobjekt repräsentiert ausführbare Dateien. Ausführbare Dateien (Programme) besitzen unter OS/2 die Erweiterungen .EXE oder .CMD. DOS- und Windows-Programme können die Erweiterungen .EXE, .COM oder .BAT haben.

Da OS/2 zuerst die Objektart ermittelt, können Programmobjekte auch nach Änderung der Erweiterung gestartet werden. Eine Änderung der Dateinamenserweiterung von ausführbaren Dateien ist aber im allgemeinen nicht sinnvoll. Haben Sie nämlich OS/2 einmal davon überzeugt, daß die ihm vorliegende Datei kein Programm ist, wird es Ihnen später schwer fallen, das System vom Gegenteil zu überzeugen.

4.2.4 Das Einheitenobjekt

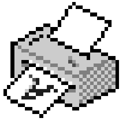


Abb. 4.4: Das Einheitenobjekt

Objekte, die eine physikalische Einheit repräsentieren, sind Einheitenobjekte. Dies sind zum Beispiel ein Druckerobjekt, ein Laufwerksobjekt, das Energieobjekt (Advanced Power Management), etc. wobei das Druckerobjekt die Verbindung zwischen Druckertreiber, Drucker und Spooler herstellt. Diese Objekte sind nicht als Datei auf der Festplatte vorhanden, sondern werden vom System in Verbindung mit Treibern im allgemeinen bereitgestellt.

4.3 Der Umgang mit Objekten

4.3.1 Objekte markieren

Um ein Objekt zu markieren (hervorzuheben oder auszuwählen), klicken Sie mit der linken Maustaste einmal darauf. Für mehrere Objekte drücken sie die [Strg]-Taste, während Sie auswählen. Sie können auch einen Rahmen um diese Objekte ziehen, oder diese mit gedrückter linker Maustaste ansteuern. Möchten Sie zu den bereits markierten Objekten welche hinzufügen oder entfernen, halten Sie die [Strg]-Taste gedrückt, während Sie die linke Maustaste betätigen. Um eine Auswahl zurückzunehmen, klicken Sie einfach mit der linken Maustaste auf eine freie Fläche des Ordners oder drücken Sie die Leertaste.

Wichtig ist, daß Sie nur Objekte innerhalb *eines* Ordners markieren können.

4.3.2 Drag&Drop

Durch »Ziehen und Übergeben« wird ein Objekt in der Anzeige bewegt. Dies geschieht durch Bewegen der Maus mit gedrückter rechter Maustaste. Sobald sie die Maustaste loslassen, wird das Objekt an der Stelle, an der sich der Mauszeiger befindet, übergeben. Was bei der Übergabe geschieht, hängt sowohl vom Objekt, das gezogen wird, als auch vom Objekt, auf das es übergeben wird, ab. Ein paar Beispiele verdeutlichen das: Ziehen Sie ein Dokument auf ein Druckerobjekt, wird dieses ausgedruckt. Ziehen Sie es auf den Reißwolf, wird es gelöscht. Ziehen Sie es in einen Ordner, wird es dorthin verschoben. Halten Sie dabei die [Strg]-Taste gedrückt, wird dort eine Kopie vom Dokument erstellt.

4.3.3 Kontextmenü

Jedes Objekt verfügt über ein Kontextmenü, in dem die momentan für das Objekt verfügbaren Funktionen (und nur diese) aufgeführt sind. Dabei handelt es sich um eine Liste von Befehlen, die auf das Objekt angewendet werden können. Um das Kontextmenü eines Objekts zu erhalten, genügt ein Mausklick mit der rechten Maustaste darauf, oder über die Tastatur mit der Tastenkombination [Shift]+[F10], wenn es ausgewählt wurde. Es ist dann grau hinterlegt. Bei Ordnern genügt auch die Betätigung der rechten Maustaste in einer freien Fläche des zugehörigen Fensters.

Sind mehrere Objekte markiert, listet deren Kontextmenü nur die gemeinsamen Bearbeitungsmöglichkeiten auf.

4.3.4 Öffnen

Zum Öffnen eines Objekts klicken Sie mit der linken Maustaste zweimal hintereinander (Doppelklick) darauf, oder wählen Sie aus dessen Kontextmenü den Menüpunkt *Öffnen*. Wenn ein Objekt geöffnet ist, wird es mit einer Schraffur gekennzeichnet.

Grundsätzlich reicht es aus zu wissen, daß mit der linken Maustaste Objekte ausgewählt und mit der rechten Maustaste Objekte manipuliert werden. Später werde ich zeigen, wie man die Funktionen der Maustasten auch ändern kann.

Im weiteren Verlauf dieses Kapitels sollen Mausaktionen, falls nicht anders erwähnt, der linken Maustaste zugeordnet werden.

4.4 Die Objekte der Arbeitsoberfläche

Auf ihrer Arbeitsoberfläche, die ein Ordner mit besonderen Eigenschaften ist, befinden sich direkt nach der Installation die Objekte *System*, *Unterstützung*, *Verbindungen*, *Programme* und *Reißwolf*. Zusätzlich befinden sich hier noch das URL-Objekt *Netscape Navigator* und der Ordner *WebExplorer*, falls TCP/IP installiert wurde.



Abb. 4.5: Die Arbeitsoberfläche und der Ordner *System*

4.4.1 Das Fenster und seine Eigenschaften

Am Fenster des Ordners *System*, das Sie nach dem Öffnen erhalten, wollen wir uns mit den Fensterelementen eines Standardfensters auseinandersetzen. Ein Fenster ist ein abgegrenzter Bereich auf dem Bildschirm, in dem Informationen angezeigt und bearbeitet werden. Prinzipiell besteht ein Fenster aus Rahmen, Titelleistensymbol, Titelleiste, den Knöpfen für Schließen, Symbolgröße/Verdecken, Maximieren/Wiederherstellen, der Menüleiste, der Schiebeleisten und natürlich dem Arbeitsfeld. Der Rahmen umgrenzt die Fensterfläche und dient auch dazu, sie durch Ziehen mit der Maus zu vergrößern oder zu verkleinern. Das Titelleistensymbol ruft das Kontextmenü des aktiven Fensters auf und interpretiert einen Doppelklick als Schließ-Befehl. Die Hauptfunktion der Titelleiste ist es, über den Titel des jeweiligen Objektes zu informieren. Mit ihr kann man aber zusätzlich ein Fenster verschieben oder durch einen Doppelklick maximieren und wiederherstellen. Über sie erkennt man auch, ob ein Fenster aktiv ist oder nicht, sprich, ob unsere Tastatureingaben in diesem Fenster erfolgt oder nicht. Das aktive Fenster wird farblich von den anderen hervorgehoben. Der Knopf zum Schießen schließt, wie der Name schon sagt, die Sicht auf das jeweilige Objekt. In der objektorientierten Welt spricht man von Ansichten (Darstellungen) der Objekte beziehungsweise

Daten. Der Knopf für Symbolgröße/Verdecken bewirkt, daß ein Fenster aus dem sichtbaren Bereich entfernt und je nach Einstellung verdeckt bzw. minimiert wird.

Die Menüleiste, neu hinzugekommen in OS/2 Warp Version 4, enthält nichts anderes als das Kontextmenü des jeweiligen Objektes, in Kategorien gruppiert, und das Kontextmenü des im Ordner ausgewählten Objektes.

Die Schiebeleisten erscheinen, sobald sich ein Objekt in einem Fenster außerhalb des sichtbaren Fensterrahmens befindet. Das Arbeitsfeld ist die vom Fensterrahmen eingeschlossene Fläche.

4.5 Die Fensterliste



Abb. 4.6: Die OS/2-Fensterliste

Um den Überblick über die einzelnen Fenster zu bewahren und zwischen ihnen umschalten zu können, gibt es in OS/2 die Fensterliste. Es handelt sich bei ihr um ein besonderes Fenster, in dem alle Fenster, in der Reihenfolge, mit der sie geöffnet wurden, aufgelistet werden. Sie erhalten die Fensterliste auf drei Arten:

– über die Tastenkombination [Strg]+[Esc]

– indem Sie beide Maustasten auf einer freien Fläche der Arbeitsoberfläche betätigen.

– über das Kontextmenü eines Festers, indem Sie Fensterliste beziehungsweise *Fenster|Fensterliste* wählen

Weitere Möglichkeiten, zu den geöffneten Objekten zu gelangen sind:

– Die Tastenkombination [Alt]+[Esc]. Damit bewegen Sie sich von Fenster zu Fenster, wobei die minimierten Fenster ignoriert werden.

– Die Tastenkombination [Alt]+[Tab]. Sie dient als Umschalter (Switcher) zwischen den jeweiligen geöffneten, nicht minimiert auf der WPS laufenden Fenstern. Je nach Objekteinstellungen wird ein minimiertes Objekt im Ordner *System|Fenster* in Symbolgröße abgelegt oder auf der Arbeitsoberfläche als Icon dargestellt. Wichtig ist, daß OS/2 Ihnen beim Umschalten zwischen den Fenstern die Möglichkeit bietet, ein nicht auf Systemanfragen reagierendes Fenster zu schließen. Dabei kann der Aufruf der Fensterliste einige Sekunden bis zu circa zwei Minuten dauern. Wenn aber auf Grund eines abgestürzten Programms, das sich im Fokus befindet, die Tastatureingaben nicht verarbeitet werden können oder der Tastaturpuffer voll ist, können Sie ihn mit der Tastenkombination [Strg]+[Pause] leeren und dann durch [Strg]+[Esc] die Fensterliste aufrufen. Über den Eintrag *Tasklist* in der Umgebungsvariable »Autostart« der »config.sys« können Sie die Fensterliste aktivieren oder deaktivieren.

4.6 Die Objekt-Einstellungen

Jedes Objekt besitzt ein Kontextmenü, das sich den Objektmöglichkeiten anpaßt. Das Kontextmenü jeden Objekts enthält den Punkt *Einstellungen*. Die Möglichkeit, ein Objekt Ihren Bedürfnissen anzupassen, ist ein Teil des objektorientierten Konzepts.

Diese Möglichkeiten werden in einem karteikastenähnlichen Fenster aufbewahrt und verwaltet. Dieser ist in Fächer aufgeteilt, die wiederum mehrere Seiten enthalten können. Sie können ein Fach durch einfachen Mausklick auf sein Register (Titel, Indexzunge), oder über ein Pop-up-Menü, das Sie mit einem Klick der rechten Maustaste erhalten, erreichen. Für diesen Karteikasten wird öfter auch der Begriff (Einstellungs-)Notizbuch verwendet, da bis zur Version 3 dessen Erscheinungsbild einem Notizbuch glich. Jede Basisobjektklasse besitzt natürlich Möglichkeiten, die für sie spezifisch sind, wobei es auch Gemeinsamkeiten gibt.

4.6.1 Der Ordner

Der Einstellungs-Karteikasten des Ordners enthält folgende Fächer:

Symbol, *Fenster*, *Spracheingabe* (falls Voice Type installiert wurde), *Datei*, *Menü*, *Hintergrund*, *Anzeige*, *Sortieren* und *Anzeigekriterien*.



Abb. 4.7: Einstellungskarteikasten eines Ordners

Symbol

Auf der ersten Seite des Fachs *Symbol* können Sie den Titel des Ordners, sein Sinnbild (Icon) (bei geschlossenem Zustand) verändern, sowie ihn zur Schablone erklären und ihn auf seiner Position innerhalb des Ordners, fixieren. Eine Schablone ist nichts anderes als eine Vorlage für ein bestimmtes Objekt, das man sich als Haftnotizblock vorstellen kann. Auf die Schablone wird später noch genauer eingegangen. Um das Icon zu ändern, reicht es auch aus, per Drag& Drop« ein anderes Icon auf den »aktuellen Icon« zu ziehen. Die zweite Seite ist für das sogenannte »Animations-Icon« zuständig, das zu sehen ist, sobald der Ordner geöffnet wurde.

Fenster

Das Fenster-Fach entscheidet, ob das jeweilige Fenster ein Minimierungs-Symbol oder ein Symbol zum Verdecken hat. Durch *Verdecken* verschwindet das Fenster aus dem sichtbaren Bereich und ist dann nur über die Fensterliste und natürlich über sein eigenes Icon erreichbar. Beim Minimieren kann es anders aussehen. Wählen Sie nämlich den Punkt *Ordner|Fenster in Symbolgröße* wird beim Minimieren das Fenster dort abgelegt. Sie können natürlich auch *Auf der Arbeitsoberfläche* wählen, wobei dann die minimierten Fenster als Icons auf der Arbeitsoberfläche zu sehen sind. Haben Sie ein Fenster schon geöffnet, entscheidet das Feld *Vorgang beim Öffnen des Objekts,* ob beim erneuten Öffnen dieses nochmal erstellt wird oder ein Sprung zum bereits offenen Fenster erfolgt. Die zweite Seite in diesem Fach dient der Ordnung und der Übersicht auf der Arbeitsoberfläche.

Mit dem Feld *Ordner automatisch schließen* können Sie einstellen, ob der Ordner geschlossen wird, sobald Sie einen enthaltenen Ordner oder ein Objekt in diesem öffnen.

Datei

Das Feld *Thema* (aus dem engl. »subject«) ist zur Eingabe einer Beschreibung des Objekts gedacht. Leider wird es in der Praxis genauso oft verwendet, wie dies auch IBM tut – nämlich überhaupt nicht. Im Feld *Physikalischer Name* werden der Pfad und der Name des Objekts angezeigt, da sich der Titel eines Objekts auf der WPS von seinem Namen auf der physikalischen Einheit unterscheiden kann. Auf diese Weise haben Sie als Anwender keinerlei Einschränkungen bei der Namensvergabe ihrer Objekte. Somit sind auch Objekttitel wie z.B.: »\ / :-) & %?*« unter OS/2 kein Problem.

Das Markierungsfeld *Arbeitsordner* hilft Ihnen, dort mit ihrer Arbeit weiterzumachen, wo Sie das letzte Mal aufgehört haben. Es bewirkt, daß mit dem Öffnen eines Arbeitsordners auch alle Objekte geöffnet werden, die sich in ihm befinden und geöffnet waren, als dieser geschlossen wurde. Ein Arbeitsordner hat aber noch eine weitere Eigenschaften. Wird er verdeckt, werden die in ihm geöffneten Objekte auch verdeckt. Wird dieser im Order *Fenster in Symbolgröße* minimiert, werden auch seine Objekte minimiert, wobei nur dieser im Symbolgröße-Ordner erscheint. Wichtig ist, daß er nur bei der Symbol- bzw. Detailanzeige seinen Dienst verrichtet. Auf der zweiten Seite werden folgende Informationen angegeben:

Datum/Uhrzeit: der Erstellung, der letzten Änderung und des letzten Zugriffs.

Größe: Inhalt (Anzahl der Objekte), erweiterte Attribute (Größe in Bytes).

Zusätzlich erhält der Anwender die Möglichkeit, die Attribute für das Objekt zu ändern.

Die Felder *Kommentar* und *Schlüsselwort* auf der dritten Seite stehen Ihnen zur freien Verfügung. Das Feld *Protokoll* dient zur Anzeige chronologischer Informationen, die von Programmen oder vom System bereitgestellt werden.

Menü

Im Fach *Menü* haben Sie die Möglichkeit, das Kontextmenü eines Ordners zu ändern oder auch zu erweitern. Es besteht aus zwei Feldern. Das eine listet die verfügbaren Menüs (die Hauptmenüpunkte) und das andere die zugehörigen Aktionen im Menü (die Untermenüpunkte) auf. Sie können das Kontextmenü mit Programmobjekten erweitern. Diese können entweder als Hauptmenüpunkt oder in Untermenüs verschachtelt dargestellt werden. Wollen Sie ein Programm hinzufügen, brauchen Sie (während in der Liste der Menüs das primäre Kontextmenü gewählt ist) dieses nur dem Feld *Aktionen im Menü* per Drag&Drop zu übergeben. Wollen Sie hingegen einen ganzen Untermenüpunkt erstellen, wie zum Beispiel »Utilities«, wählen Sie *Erstellen* in der Liste der Menüs und geben den Namen ein. Hierbei bewirkt eine Tilde »~« vor einem Buchstaben des Menünamens, daß dieser Buchstabe zum Tastaturkürzel für diesen Menüpunkt wird. Sie können zwischen *Untermenü* und *Bedingtes Untermenü* als Menütyp wählen. Dabei bewirkt das bedingte Untermenü nur, daß Sie ein Programm aus diesem Menü bei dessen Aufruf als Standardaktion wählen können. Dies passiert zum Beispiel beim Menüpunkt *Öffnen*, bei dem als Standard die *Symbolansicht* markiert und ausgeführt wird.

Nun aber zur neu hinzugekommenen zweite Seite. Hier können Sie die Kontextmenüs der Objekte in einem Ordner verkürzen, eine Funktion die auch bei OS/2 Warp Version 3 über Umgebungsvariablen für die WPS erreichbar war. Zusätzlich können Sie auch die Menüleiste des Ordners deaktivieren.

Hintergrund

Falls Ihnen die Standardhintergrundfarbe der Ordner etwas zu blaß erscheint, sind Sie in diesem Fach genau richtig, um dies zu ändern. Dabei können Sie eine andere Farbe, eine Bitmap oder auch beides zur Veränderung des Hintergrunds einsetzen. Falls Sie eine bestimmte Bitmap oder eine Farbe aus den Farbpaletten zum Hintergrund machen möchten, brauchen Sie diese einfach nur auf das Feld *Bildanzeige* zu ziehen. Sie können natürlich auch eine Farbe aus dem Farbkreis wählen, der sich hinter dem Druckknopf *Farbe ändern...* verbirgt. Der Farbkreis bietet Ihnen auch die Möglichkeit, eine Farbe duch Eingabe ihrer Farbwerte (RGB/FSH) zu definieren.

Beim Pull-down-Menü *Bild* können Sie eine Bitmap auswählen. Dort werden diejenigen Bitmaps angezeigt, die sich im Verzeichnis »*Systemlaufwerk*:\OS2\Bitmaps« befinden und alle die Bitmaps, die Sie irgendwann einmal benutzt haben.

Anzeige

Dieses Fach besteht aus drei Seiten, wobei jede Seite für eine der drei Darstellungsmöglichkeiten der Objekte eines Ordners zuständig ist. Im Fach Menü haben Sie gesehen, daß für den Vorgang des Öffnens die Symbolanzeige als Standardfunktion eingestellt ist. In der Symbolanzeige können Sie wählen, ob die Icons (Symbole) frei, in Gitterform (gerastert) und ein- oder mehrspaltig plaziert werden sollen. Sie haben auch die Möglichkeit, die Icons zu verkleinern oder gar unsichtbar zu machen. Letzteres steht Ihnen nur bei den spaltenförmigen Anordnungen zur Verfügung. Sind ihre Icons so aussagekräftig, daß Sie völlig auf den Titel der Objekte verzichten möchten, reicht ein Mausklick auf das Markierungsfeld *Sichtbar*, um ihren Wünschen nachzukommen. Die Strukturanzeige stellt den Inhalt eines Ordners in Form eines Verzeichnisbaumes dar. Auf der zweiten Seite können Sie nun wählen, ob dieser mit oder ohne Linien angezeigt werden soll. Brauchen Sie zu den Objekten eines Ordners mehr Informationen, dann ist die Detailanzeige für Sie das richtige. Die Einstellungen dazu befinden sich auf der dritten Seite. Sie können dort im Feld *Objekttyp* erstmal den Typ aussuchen, über den detaillierte Informationen angezeigt werden sollen. Danach wählen Sie aus den möglichen Informationen die, die Sie benötigen. Nehmen wir an, Sie möchten von den Datendateien eines Ordners die jeweilige Größe wissen. Dazu wählen sie den Typ *Datendatei* aus und deselektieren alle Details außer der *Größe*. Öffnen Sie nun die Detail-Anzeige dieses Ordners, wird in der Spalte *Größe* nur dann etwas angezeigt, wenn das jeweilige Objekt eine Datendatei ist.

Sortieren

Objekte in einem Ordner können nach unterschiedlichen Kriterien sortiert sein. Dabei kann für das Ordnerobjekt auch festgelegt werden, ob die gewählte Sortierfolge immer beibehalten werden soll, oder ob der Sortiervorgang erst nach Aufforderung stattfinden soll. Sie haben die Möglichkeit, den Objekttyp, nach dem sortiert werden soll, zu wählen. Die erlaubten Sortierattribute werden dem Kontextmenü hinzugefügt.

Anzeigekriterien

Hierüber können Sie, nach verschiedenen Kriterien, Objekte von der Anzeige ausschließen. Das heißt, daß sie zwar vorhanden, aber nicht sichtbar sind. So können Sie für den Ordner *Laufwerke* einstellen, daß das Objekt *Laufwerk B* nicht angezeigt wird. Die Anwendungsmöglichkeiten für die Anzeigekriterien sind so vielfältig, daß jeder von Ihnen sicherlich etwas damit anfangen kann. Sei es auch in Verbindung mit der Detailanzeige, nur die Überprüfung der Größe der Auslagerungsdatei.

4.6.2 Das Programmobjekt

Das Programobjekt repräsentiert eine ausführbare Datei (auch Programm genannt). Ein Programm ist also eine Datei, die eine Folge von Instruktionen enthält, die der Computer verstehen und interpretieren kann. Ein Öffnen dieser Datei startet das Programm beziehungsweise es führt der Computer die in der Datei enthaltenen Instruktionen aus. Sie sollten die Begriffe Programm(datei)objekt bzw. Programm und Programmdatei nicht gleichsetzen. Für eine Programmdatei können Sie mehrere Programm(datei)objekte erstellen. Änderungen, die Sie an einem Programmobjekt vornehmen, betreffen nur dieses und nicht die zugehörige Programmdatei. Bei der Programmdatei handelt es sich um ein besonderes Datendateiobjekt, dessen Öffnung die Ausführung des Programms veranlaßt.

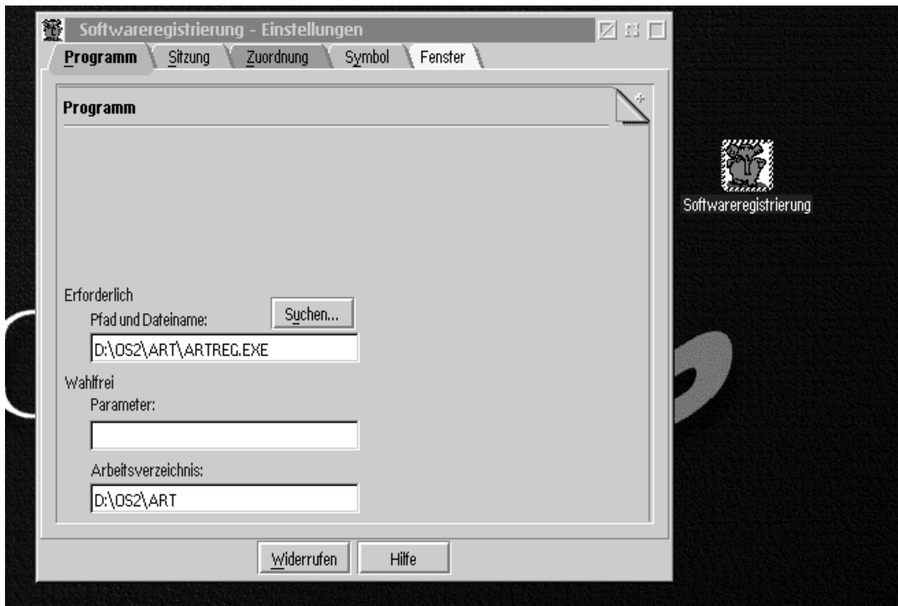


Abb. 4.8: Einstellungskarteikasten eines Programmobjektes

Der Einstellungs-Karteikasten eines Programmobjektes enthält die Fächer *Programm, Sitzung*, *Zuordnung*, *Icon* und *Fenster*.

Programm

Hier werden die für den Programmaufruf nötigen Informationen eingetragen. Sobald Sie das Feld *Pfad und Dateiname* verlassen, wird Ihre Eingabe mit einer Datenbank verglichen, in der für mehrere tausende Programme die optimalen Einstellungs-Informationen enthalten sind. Wird Ihr eingegebenes Programm dort endeckt, nimmt OS/2 diese Einstellungen automatisch vor. Im Feld *Parameter* können Sie Ihre gewünschten Startparameter für das Programm eingeben. Haben sie ein Programm, das zwar Startparameter benötigt, diese unterscheiden sich aber von Mal zu Mal, geben sie einfach »[ ]« in das Parameterfeld ein. Beim Starten des Programmobjekts erhalten Sie nun über ein Dialogfenster die Möglichkeit, die erforderlichen Parameter einzugeben. Innerhalb der eckigen Klammern können Sie auch einen Text eintragen, der dann im Dialogfenster erscheint.

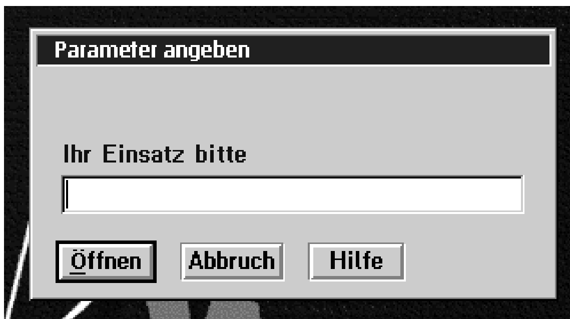


Abb. 4.9: Das Dialogfenster

Weitere von OS/2 bereitgestellte Parameter sind:

– %**D für das Laufwerk

– %**F für den Dateinamen

– %**N für den Dateinamen ohne Erweiterung

– %**E für die Erweiterung (ohne Punkt)

– %**P für das Laufwerk mit Pfad (ohne abschließenden Backslash)

– %* für den kompletten Dateinamen mit Pfad

– % für keinen Parameter

– ? für den Aufruf einer Dialogbox zur Parametereingabe

Falls Sie eine Datei auf ein Programmobjekt per Drag&Drop ziehen, wird standardmäßig der vollständige Pfad mit dem Dateinamen dem Programm übergeben.

Sitzung

Handelt es sich nicht um ein Presentation-Manager- (PM) bzw. WPS-Programm, kann die Umgebung (z.B. OS/2-, DOS-, Windows-Fenster bzw. Gesamtbildschirm) ausgewählt werden, in der es ausgeführt wird. Dabei wird bei OS/2-Programmen der Befehlsinterpreter »cmd.exe« aufgerufen. Zur DOS- und Windows-Umgebung lesen Sie das zugehörige Kapitel in diesem Buch.

Zuordnung

In diesem Fach können Sie bestimmen, welche Dateitypen das Programm bearbeiten kann. Das Programm wird dann bei den gewählten Dateitypen im bedingten Untermenü *Öffnen* dem Kontextmenü hinzugefügt. Sie können auch mehrere Dateinamen eingeben, für die dieses Programm zur Verfügung stehen soll. Sie könnten z.B. alle Dateien mit der Erweiterung »txt« (*.txt), »cmd« (*.cmd), und so weiter Ihrem Lieblingseditor zuweisen.

Die Fächer *Allgemein* und *Fenster* entsprechen denen aus dem Ordnerobjekt. Natürlich sind die ordnerspezifischen Seiten bei den Programmobjekten nicht vorhanden.

4.6.3 Das Datendateiobjekt

Typ

Das einzige, was Sie im Einstellungs-Karteikasten der Datendateiobjekte nicht kennen, ist das Fach *Typ*. Dort können Sie der Datendatei einen Typ zuordnen. Sie können natürlich auch einen neuen Typ anlegen, den Sie dann einem bestimmten Programmobjekt zuordnen. Sobald eine Datendatei einem Typ zugeordnet ist, für dessen Verarbeitung kein Programmobjekt existiert, werden beim Öffnen der Datei deren Einstellungen angezeigt. Standardmäßig ist jede Datei von Typ *Plain Text*. Sprich, sie wird mit dem »OS/2 Systemeditor« geöffnet, es sei denn, sie ist schon einem anderen Programm zugewiesen.

Bei Programmdateien kann man den fließenden Übergang von Objektklassen beobachten. Sie besitzen nämlich alle Einstellungsmöglichkeiten einer Datendatei, wobei ein Öffnen dieser Datei die Ausführung des Programms bewirkt. Über ihr Kontextmenü können Sie für diese Programmdatei ein »Programmobjekt« erstellen. Gleiches erreichen Sie, wenn Sie eine Programmdatei ziehen.

Das nun zu den Basisobjektklassen. Die Detailanzeige eines Ordners hilft Ihnen, mehr über die verschiedenen Objektklassen zu erfahren.

4.7 Objekte der Arbeitsoberfläche

Nun zu den Objekten der WPS. Diese werden Ihnen bei der Arbeit und bei der Konfiguration von OS/2 weiterhelfen. Auf der WPS befinden sich direkt nach der Installation folgende Objekte:

4.7.1 Der Ordner *System*

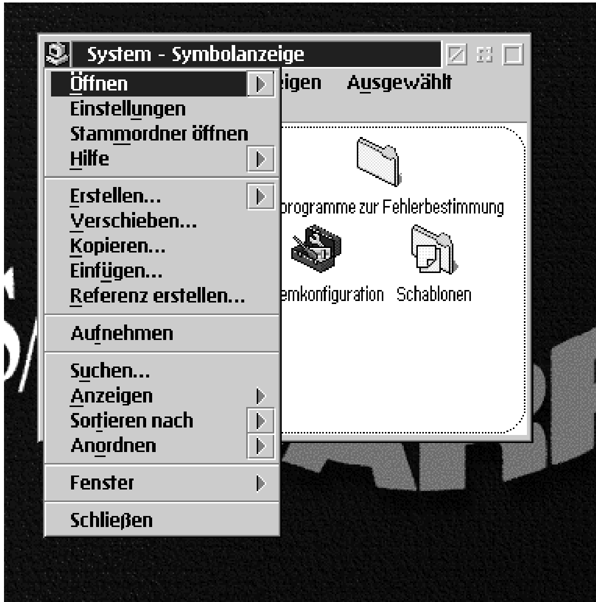


Abb. 4.10:
Der Systemordner

Hier handelt es sich um einen »normalen« Ordner, dessen Objekte in direktem Zusammenhang mit dem Betriebssystem stehen. In ihm befinden sich unter anderem die Befehlszeilenobjekte (Shells, Prompts), die Laufwerksobjekte und die Objekte zur Systemkonfiguration und Fehlerbehandlung. Das Kontextmenü dieses Ordners listet die für diesen Ordner möglichen Befehle auf. Diese passen sich dem Zustand des Ordners an. Ist dieser geöffnet, erweitert sich das Kontextmenü um Befehle, die zur Bearbeitung der in ihm enthaltenen Objekte dienen. Bei geschlossenem Ordner stehen folgende Befehle zur Verfügung:

Öffnen

Dieses bedingte Untermenü dient zum Aufruf einer Objektansicht. Für den Ordner sind dies die Symbolansicht, die Struktur- und die Detailanzeige, auf die ich schon bei den Basisobjektklassen eingegangen bin.

Einstellungen

Damit erreichen Sie den Einstellungs-Karteikasten für diesen Ordner.

Hilfe

OS/2 besitzt eine kontextsensitive Hilfe. Das bedeutet, daß Sie die Hilfeinformationen angezeigt bekommen, die für das Objekt, von dem aus die Hilfe aufgerufen würde, verfügbar sind. Sie können jederzeit mit der [F1]-Taste die Hilfefunktion aufrufen. Eine umfangreiche Sammlung an Hilfedateien defindet sich im Ordner *Unterstützung*.

Erstellen

Ein gerne übersehener Befehl, mit dem Sie ein Objekt mit der gleichen Klasse wie des vorliegenden erstellen können. Über die Untermenüpunkte können Sie zwischen *Standard*, *Ordner* und *Neuer Ordner* wählen. Dabei steht:

– S*tandard* für Betriebssystemeinstellungen

– *Ordner* für die Vorlage im Schablonenordner

– *Neuer Ordner* für die im Arbeitsoberflächenordner *Programme* befindliche Schablone. Somit erhält das neu erstellte Objekt die Einstellungen der jeweiligen Vorlage. Wird einer der Punkte aufgerufen, erscheint ein Fenster, in dem Sie den neuen Objektnamen und den Zielort des Objekts definieren können.

Verschieben und Kopieren

Beim *Verschieben* und *Kopieren* wird das Objekt mit seinem gesamten Inhalt verschoben bzw. kopiert. Beim Verschieben kommt die Objektorientiertheit von OS/2 sehr deutlich zum Ausdruck. Wird ein Objekt verschoben, werden alle mit ihm in Verbindung stehenden Objekte aktualisiert.

Einfügen

Dieser Punkt erscheint nur dann, wenn sich in der Zwischenablage (Clipboard) Daten befinden. Er bietet Ihnen die Möglichkeit, von diesen Daten ein Objekt zu erzeugen. Dabei werden automatisch das Format (Typ) und die Objektklasse der Daten ermittelt und Ihnen für die neue Datei vorgeschlagen.

Referenz erstellen

Gute Referenzen braucht man fürs Leben. Eine Referenz (engl. »Shadow«) ist ein Verweis auf ein Objekt und stellt ein Objekt samt seiner Funktionalität dar. Der Sinn von Referenzen besteht darin, Ihnen ein Objekt in mehreren Ordnern zur Verfügung zu stellen. Änderungen, die Sie an einer Referenz vornehmen, betreffen das Originalobjekt. Sie erkennen Referenzen über ihre etwas hellere Titelfarbe und den Kontextmenüpunkt *Original*. Darüber können Sie das Originalobjekt *Suchen*, *Kopieren* und – falls möglich – *Löschen*. Referenzen lassen sich immer löschen. Dieser Löschvorgang wirkt sich nicht auf das Originalobjekt aus.

Aufnehmen

Diese Funktion soll ihnen den Drag&Drop-Vorgang erleichern.

Sie können Objekte aus einem Ordner aufnehmen und zu einem späteren Zeitpunkt übergeben. Sobald der Ordner, aus dem aufgenomen wurde, geschlossen wird, bricht der Aufnahmevorgang ab.

Suchen

Hinter diesem Punkt verbirgt sich, wie Sie sicherlich angenommen haben, eine Suchfunktion. Ihre Leistungsfähigkeit tritt in Erscheinung, sobald Sie den Button *Weiter..*. betätigen. Sie können nämlich zusätzlich zum Objekttitel auch andere Kriterien für die Suche definieren. Im Feld *Ordner* können Sie die Lokation in Form eines Pfads eingeben, oder über *Lokalisieren* diesen auswählen. Wird die Suche über das Kontextmenü eines Ordners aufgerufen, wird dessen Position als Anfangspunkt eingetragen. Sie erweitern den Suchpfad über das Markierungsfeld *Enthaltene Ordner einbeziehen*. Es sollte beachtet werden, daß die Suche dem physikalischen Pfad folgt, wobei jegliche Objekte erkannt werden. Das bedeutet: Wenn Sie auf der Arbeitsoberfläche (inklusive enthaltener Ordner) suchen lassen, werden auch die auf ihr befindlichen Referenzordner erkannt. Deren Inhalt wird aber nur dann durchsucht, wenn sich das Originalobjekt innerhalb der Arbeitsoberflächen-Verzeichnisstruktur befindet.

Symbol fixieren

Wie schon erwähnt, wird damit ein Objekt innerhalb eines Ordners auf seiner Position fixiert. Dadurch haben Sortier- und Anordnungsbefehle für den Ordner, in dem sich das fixierte Objekt befindet, keine Wirkung auf das Objekt.

Wie oben erwähnt, passen sich die Befehle im Kontextmenü dem Zustand des Ordners an. Wird dieser nun geöffnet, erweitert sich die Befehlspalette um folgende Optionen:

Stammordner öffnen

Dieser Befehl bewirkt das Öffnen des Stammordners, in dem sich der Ordner befindet, der zur Zeit bearbeitet wird. Unter Stammordner ist der in der Verzeichnisstruktur eines Laufwerks übergeordnete Ordner zu verstehen.

Anzeigen

Hierüber können Sie zwischen den verschiedenen Anzeigearten wechseln. Desweiteren finden Sie die Menüpunkte *Alles auswählen*, *Auswahl zurücknehmen* und *Sofort aktualisieren*. Diese Befehle dienen der Objektbearbeitung im Ordner.

Sortieren nach

Dieser Menüpunkt dient zur Sortierung der im Ordner enthaltenen Objekte. Die Sortierfolge kann festgelegt werden.

Anordnen

Heißt es nicht, »Ordnung ist das halbe Leben«? Nun haben Sie genügend Anordnungsmöglichkeiten. Sie können die Objekte eines Ordners von oben, links, rechts, unten, horizontal, vertikal und standardmäßig bezüglich der sichtbaren Fensterfläche anordnen lassen. Dabei ist zu beachten, daß die vertikale beziehungsweise horizontale Anordnung für die markierten Objekte erfolgt. Haben Sie den Befehl gewählt, ist die Angabe eines Bezugspunkts im Ordner erforderlich.

Anordnen/Sortieren widerrufen

Wurde einer der Befehle *Anordnen* oder *Sortieren* benutzt, haben Sie die Möglichkeit, diesen hierüber rückgängig zu machen.

Fenster

Dieser Menüpunkt ist in Verbindung mit dem Unterkapitel »Umgang mit Objekten/Grundlagen« selbsterklärend.

4.7.2 Objekte im Ordner *System*

Befehlszeilen

Hier befinden sich die Kommandozeilenfenster. Es handelt sich hierbei um Programmobjekte, bei denen im Feld *Programmname* nur ein »*« steht, was für die in der »config.sys« angegebenen Befehlsinterpreter gilt. Über die Sitzungsseite wird eindeutig, wofür »*« steht. Befehlsinterpreter verarbeiten, wie der Name schon besagt, Befehle. Die für den OS/2-Kommandointerpreter »cmd.exe« verfügbaren Befehle entnehmen Sie bitte den Online-Referenzhandbüchern im Ordner *Unterstützung|Information|Referenz und Befehle*. Die WPS merkt sich bei fast allen Fenstern ihre Größe und Position auf der Arbeitsoberfläche. »Bei fast allen« deswegen, weil dies bei den Befehlszeilen nicht der Fall ist. Möchten Sie, daß ein solches Fenster immer an einer festen Position erscheint, können Sie es zur gewünschten Position verschieben. Um es dort zu fixieren, muß während des Verschiebeforganges die [Shift]-Taste gehalten werden. Entsprechendes gilt für die Fenstergröße. Zur Schriftgröße findet man im Kontextmenü eines DOS- bzw. OS/2-Fensters den Menüpunkt *Schriftartgröße*.

Im Kontextmenü befindet sich weiterhin der Unterpunkt *Markieren*, mit dem der Markiermodus für dieses Fenster aktiviert wird. Dadurch erhalten Sie die Möglichkeit, einen Fensterausschnitt in die Zwischenablage (Clipboard) zu kopieren. Standardmäßig können Sie den Markiermodus über einen einfachen Mausklick (innerhalb des Fensters) aktivieren. Dabei steht Ihnen über die rechte Maustaste auch ein Kontextmenü zur Verfügung.

Der DOS- und Win-OS/2-Umgebung ist hier im Buch ein eigenständiges Kapitel gewidmet.

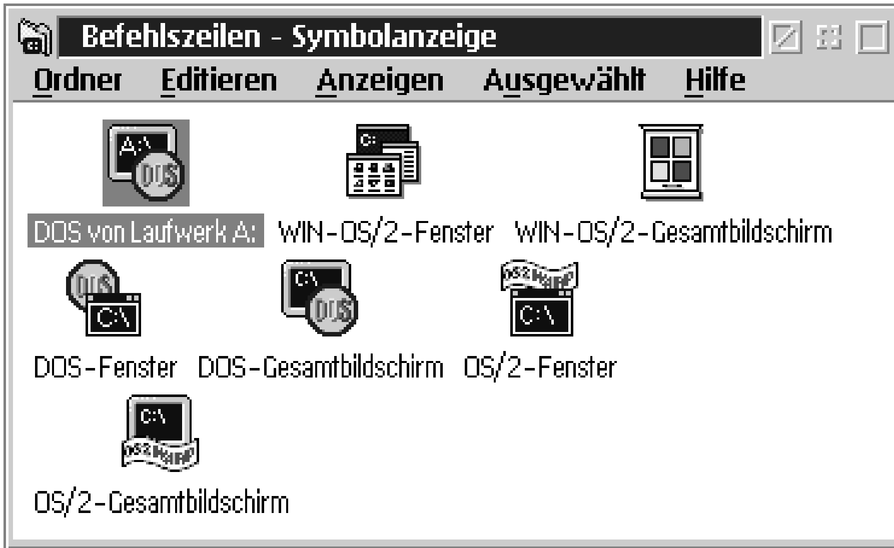


Abb. 4.11: Die Befehlszeilen

Hilfsprogramme zur Fehlerbestimmung

Hier werden Ihnen Objekte zur Fehlerbestimmung bereitgestellt. Von der Möglichkeit zur Erstellung von Speicherauszügen bis zu einem Systemfehlerprotokoll ist alles vorhanden. Es handelt sich um Utilities, die eher für einen Systemadministrator oder Programmierer als für den Normalanwender gedacht sind. Falls Sie sich damit befassen wollen, sollten Sie die Online Hilfe und entsprechende Literatur zu Rate ziehen.

Klickstartleiste

Das umstrittenste Objekt bei Warp V3 ist nun wieder da. Von den einen gehaßt, von den anderen geliebt, hat es doch den Versionssprung überlebt. Es handelt sich dabei um ein Utility zum schnellen Zugriff auf Objekte und Befehle. Diese werden, als Referenzobjekte, per Drag&Drop auf ihr plaziert. Über das Einstellungs-Notizbuch können Sie die Klickstartleiste leicht Ihren Bedürfnissen anpassen.

Laufwerke

Dieser Referenz-Ordner enthält die vom System erkannten Laufwerke. Das sind die Diskettenlaufwerke A und B, Festplattenlaufwerke (im lokalen System oder über Netzwerk), das CD-ROM-Laufwerk et cetera. Das Kontextmenü des Laufwerksordners bietet Ihnen die Möglichkeit, über den Menüpunkt *Partition erstellen*, das Programm » fdiskpm.exe« aufzurufen. Damit können Sie Ihre lokalen Festplatten einrichten.

Ein im Einstellungen-Notizbuch neu hinzugekommener Eintrag *Reserviert* ermöglicht es Ihnen, Laufwerksbuchstaben zu reservieren.

Je nach Laufwerkstyp enthält das Kontextmenü auch hier alle für den Datenträger (Laufwerk) zur Verfügung stehenden Befehle. Beschreibbare Medien (Disketten, Festplatten, ...) können Sie mit *Datenträger prüfen* über das Kontextmenü auf Fehler hin untersuchen. Weiterhin können Sie mit dem Befehl *Datenträger formatieren* das Medium für die Speicherung von Daten vorbereiten.

Bei Festplatten können Sie zwischen dem HPFS- (High Performance File System) und dem FAT-(File Allocation Table) Dateisystem wählen. Disketten können zusätzlich auch kopiert werden.

Wechselmedien wie zum Beispiel CD-ROM-Laufwerke oder MO-Laufwerke können Sie zusätzlich sperren (mounten), freigeben und auswerfen.

Schablonen

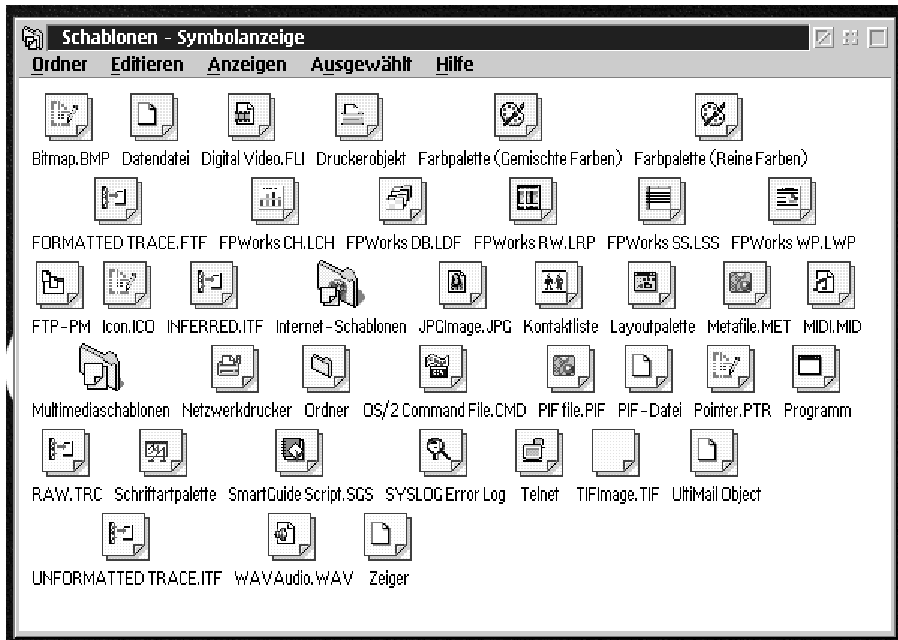


Abb. 4.12: Der Schablonenordner.

Schablonen (Templates) sind Vorlagenobjekte, die auf einfache und bequeme Weise (Drag&Drop) neue Abbilder oder Instanzen bestimmter Objekte erzeugen. Fast jedes Objekt läßt sich zur Schablone erklären. Im allgemeinen sind nur die Einheitenobjekte (z.B. Laufwerksobjekte, der Reißwolf, etc.) davon ausgeschlossen. Mit der Schablone läßt sich ein neues Objekt mit den im Schablonenobjekt vorgegebenen Einstellungen erstellen. So zum Beispiel ein nach Dateigröße sortierter Ordner, ein Formular mit Ihrem Briefkopf oder ein auf DIN A3 eingestelltes Druckerobjekt.

Neue Schablonenobjekte werden dem Kontextmenüpunkt *Erstellen* der betreffenden Objektklasse hinzugefügt.

Sie brauchen somit, um eine neue Instanz eines von Ihnen erstellten Schablonenobjektes zu erstellen, nicht unbedingt über die Schablone selbst gehen, sondern können einfach den Menüpunkt *Erstellen* eines Objektes anwählen.

Systemkonfiguration

Wie der Name schon sagt, enthält dieser Ordner alle zur Installation und Konfiguration Ihres Systems benötigten Elemente.

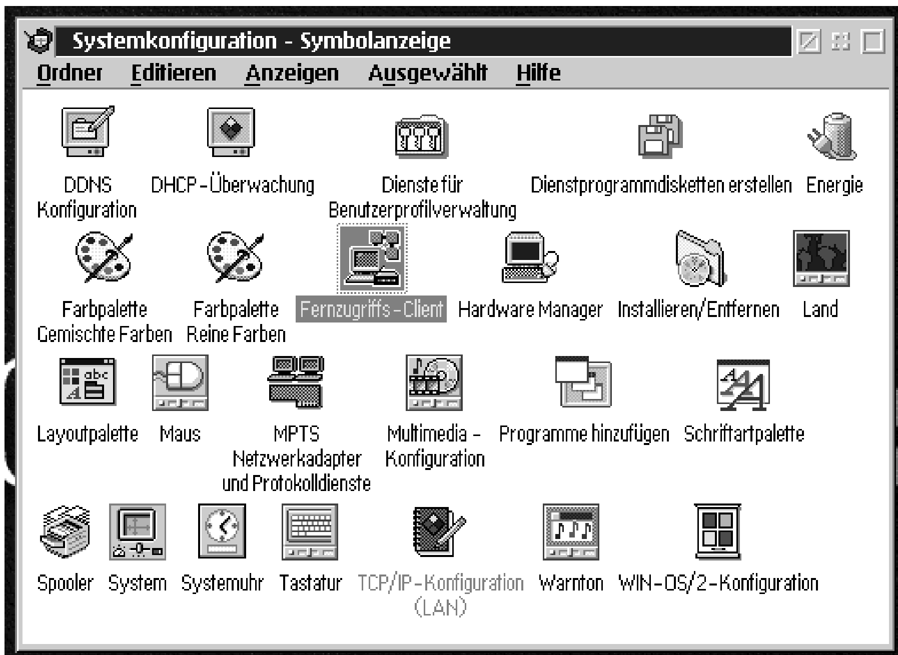


Abb. 4.13: Systemkonfigurations-Ordner

Dienstprogrammdisketten erstellen

Mit diesem Programmobjekt können Sie OS/2-Startdisketten erstellen. Diese können Sie dazu verwenden, Ihren Rechner von Diskette zu starten. Sämtliche eingebundene Treiber und Einstellungen des aktuell laufenden Systems werden hierbei übernomen.

Energie

Für den Fall, daß Ihr Rechner den Industriestandard APM (Advanced Power Management) unterstützt, haben Sie hier die Möglichkeit, den Stromverbrauch zu überwachen und den Energiesparmodus (Wartemodus) zu aktivieren.

Farbpalette Reine Farben

Hier werden die 16-Standard-VGA-Farben dargestellt. Diese können per Drag&Drop an Objekte übergeben werden. Ziehen Sie eine Farbe auf ein Ordnerfenster, wird diese Farbe als Hintergrundfarbe übernommen. Dabei bewirkt das Halten der [Alt]-Taste, daß für alle Fenster diese Farbe als Hintergrundfarbe eingestellt wird. Die Schriftartfarbe eines Ordners wird in Verbindung mit gehaltener [Strg]-Taste geändert. Durch Doppelklick auf eine Farbe gelangen Sie zu einem Farbkreis, mit dem Sie die Farbe beliebig editieren/auswählen können.

Farbpalette Gemischte Farben

Auf gleiche Art wie bei den reinen Farben können Sie hier Farben auswählen bzw. editieren. Hier stehen Ihnen unabhängig von der eingestellten Systemfarbtiefe 256 Farben zur Verfügung. Über den Kontextmenüpunkt *Erstellen* können Sie weitere Farbpaletten erzeugen.

Hardware-Manager

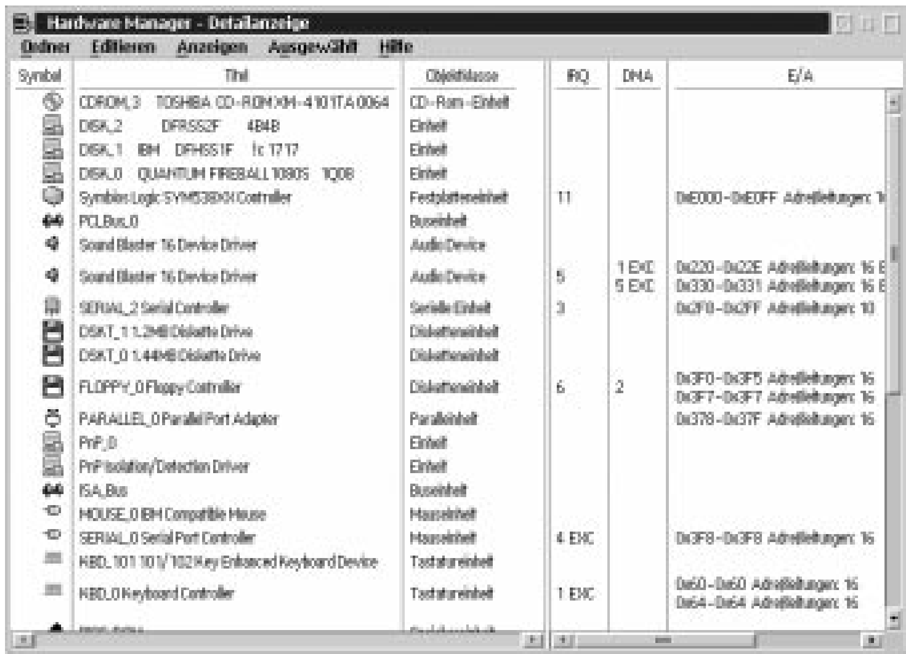


Abb. 4.14: Der Hardware-Manager

Ein altes Programm in neuem Glanz ist der Hardware-Manager. Die alten OS/2-Hasen unter Ihnen werden es vielleicht angenommen haben: Es handelt sich hierbei um den textbasierten Ressource-Manager »rmview.exe« mit einem grafischen Frontend. Doch die Hardwareerkennung hat ihre Spuren hinterlassen. So können Sie im Einstellungs-Notizbuch die »Erkennungsstufe« des Hardware-Managers angeben. Dabei stehen Ihnen fünf Stufen zur Auswahl,

die Sie für jeden bzw. nur den nächsten Systemstart aktivieren können. In diesem Ordner werden die vom System erkannte Hardware und die durch sie belegten Ressourcen angegeben.

Installieren/Entfernen

In diesem Ordner befinden sich Objekte zur Installation bzw. Deinstallation von Systemkomponenten. Diese sind im Kapitel »Installation« und »Netzwerk« näher erklärt.

Land

Mit diesem Objekt können Sie die landesspezifischen Einstellungen ändern. Dies wären Datums- und Zeit-Format, Zeichenumsetzungstabelle, Währungsformat und die Zahlendarstellung.

Layoutpalette

Diese Palette enthält eine Auswahl vordefinierter Layouts für Fenster. Hier gilt auch, daß Sie per Drag&Drop einem Fenster das gewünschte Layout übergeben können. Mit der [Alt]-Taste erreichen Sie eine systemweite Layoutänderung. Vorher separat definierte Fenster sind davon nicht betroffen, was auch für Farben gilt. Dieser Ordner bietet Ihnen auch die Möglichkeit, ein Layout nach Ihren Wünschen zu erstellen. Dabei können Sie von der Fensterrahmenbreite über die Hintergrundfarbe der Schiebeleisten bis zur Textfarbe der Titelleiste alles definieren.

Maus

Mit diesem Objekt lassen sich die Mauseinstellungen vornehmen. Wie schon anfangs erwähnt, besteht hier über das Fach *Konfiguration* die Möglichkeit, die Funktion der Maustasten von Links- auf Rechthandbedienung umzustellen. Das Fach *Mauszeiger* bietet eine Darstellung der aktiven Mauszeiger. Einzelne Mauspointer können Sie per »Drag&Drop« ändern. Es lassen sich auch komplette Gruppen einbinden, die sich jedoch als Unterverzeichnis im Ordner »Systemlaufwerk:\os2\pointers« befinden müssen. Dabei entscheidet ihr Dateiname, für welchen Systemzustand diese verwendet werden sollen. Über das Feld *Tasten* kann die Verwendung der Maustastenkombinationen für Standardaufgaben wie Ziehen, Fensterlistenaufruf und so weiter verändert werden. Weiterhin können Sie dem Mauszeiger eine Spur hinzufügen und die Maus-Sensibilität beeinflussen.

Programme hinzufügen

Mit diesem Objekt können der Arbeitsoberfläche Programmobjekte hinzugefügt werden. Dabei werden die ausführbaren Dateien auf der Festplatte gesucht und mit der Migrationsdatenbank »database.dat« verglichen. In ihr sind für mehrere tausend Programme die optimalen Einstellungen enthalten. Des weiteren wird auch die Programmkategorie berücksichtigt, so daß zum Beispiel ein Spiel im Ordner *Spiele* abgelegt wird, während ein Windows-Programm im Ordner *Windows-Programme* zu finden ist. Für den Fall, daß ein Programm nicht in der Datenbank enthalten ist, werden die Standardeinstellungen vorgenommen. Es besteht auch die Möglichkeit, eine eigene Datenbank anzulegen oder die bestehende zu erweitern.

Schriftartpalette

In ihr befindet sich eine Auswahl der im System installierten Schriftarten. Über den Punkt *Schriftart editieren* kann man dem System neue Schriftarten hinzufügen sowie die Schriftarten

der Palette ändern. Drag&Drop findet auch hier seine Anwendung. So können Sie zum Beispiel die Schriftart der Titelleiste durch »Ziehen und Übergeben« sehr leicht verändern. Mit dieser OS/2-Version können Sie zusätzlich zu den Adobe-Type-1-Schriftarten auch True-Type-Schriften einbinden.

Spooler

Der Spooler, eine zentrale Komponente im OS/2-Drucksystem, dient dazu, die Druckaufträge des Systems zu verwalten. Diese werden temporär auf der Festplatte zwischengespeichert, um das System durch einen anstehenden Druckauftrag nicht zu blockieren. Falls gewünscht, können Sie den Spooler über sein Kontextmenü deaktivieren.

Den Pfad für die Spooler-Datei können Sie auf der ersten Seite des Einstellungs-Notizbuchs angeben. Weiterhin können Sie die Systempriorität der Druckaufträge ändern und eine Druckumleitung aktivieren. Im Fach *Erweiterte Optionen* können Sie die Druckerjobanzeige und die Synchronisation der Druckertreiber (man denke an Netzwerkdrucker) genauer konfigurieren.

System

Das Fach *Bildschirm* dient zur Festlegung der Bildschirmauflösung (Seite 1) und der Bildwiederholfrequenz (Seite 2) für Grafikkarte und Monitor. Änderungen, die Sie hier vornehmen, werden beim nächsten Systemstart aktiv. Zu beachten ist, daß manche Grafikkartenhersteller (z.B. ELSA) diese Einstellungen über ein eigenes Programm vornehmen und somit in diesem Fach nur die aktuell eingestellte Bildschirmauflösung angezeigt wird. Haben Sie eine Auflösung mit nur 256 Farben gewählt, erweitert sich der Einstellungskarteikasten um das Fach *256 Farben,* in dem Sie eine Optimierung der Farbpalette aktivieren können. Dadurch nimmt OS/2 eine andere Abstufung der Farbskala vor, wodurch sich die Farbdarstellung verbessert.



Abb. 4.15:
Das Systemobjekt

Eine Anhäufung von globalen Systemeinstellungen sind in diesem Objekt enthalten.

Im Fach *Bestätigung* können Sie Systemabfragen aktivieren bzw. deaktivieren. Diese beziehen sich auf die Standardoperationen Löschen, Verschieben und Kopieren, sowie auf Änderungen der Dateierweiterung eines Objekts. Seit dieser Version können Sie auch die Sicherheitsabfrage beim Schließen von OS/2- bzw. DOS-Fenstern abschalten.

Die Funktion des Fachs *Fenster* ist Ihnen bis auf das Feld *Animation* bekannt. Sie sollten beachten, daß es sich hierbei um systemweite Einstellungen handelt. Die Animation bewirkt einen grafischen Effekt beim Öffnen eines Objekts.

Mit dem Fach *Benutzerschnittstelle* wurden Funktionen, die teilweise schon bei OS/2 Warp Version 3 über Umgebungsvariablen implementiert waren, dem Einstellungskarteikasten hinzugefügt. So können Sie das »Ziehen ganzer Fenster« aktivieren, wodurch diese bei Verschiebungen sichtbar bleiben. Mit der *Asynchronen Fokuswechsel*-Funktion weisen Sie das System an, den Fokus von einem nicht reagierenden Fenster oder Programm zu nehmen. Dieses soll helfen, eine »hängende« Message-Queue und lange Wartezeiten des Systems auf nicht reagierende Anwendungen zu vermeiden.

Mit dem Feld *Mausaktionen* wird der Markiermodus und ein Kontextmenü über die Maus in OS/2- und DOS-Fenster aktiviert. Wird das Feld *Fensterwechsel mit Alt-Tab* deaktiviert, entspricht die bereits erwähnten [Alt]-[Tab]-Tastenkombination der [Alt]-[Esc]-Kombination. Im Fach *Anzeigendruck* können Sie den Bildschirm- bzw. Fensterausdruck über die Taste [Print Screen] aktivieren/deaktivieren.

Unter *Logo* ist das bei vielen Programmen verwendete Informationsfenster beim Programmstart zu verstehen. Sie können die Dauer der »Logoerscheinung« beeinflussen, indem Sie zwischen *Logo unbegrenzt*, *Logo zeitlich begrenzt* und *Kein Logo* wählen.

Systemuhr

Wie der Name schon sagt, können Sie über dieses Element die Systemuhrzeit und das Datum aufrufen und ändern.

Tastatur

Hierüber können Sie die Tastenkombination für den Aufruf des Kontextmenüs bzw. zur Titeltext-Bearbeitung ändern. Weiterhin können Sie das Cursorblinken, die Wiederholfrequenz und Wiederholverzögerung einstellen. Über *Sonderfunktionen* soll vor allem körperbehinderten Anwendern geholfen werden, die nicht in der Lage sind, mehrere Tasten gleichzeitig zu betätigen.

Warnton

Grundeinstellungen für die Systemtöne werden hier vorgenommen. Haben Sie eine Soundkarte installiert, können Sie hier die Systemtöne einstellen, wobei Ihnen komplette Klangschemata zur Verfügung stehen.

Multimedia-Konfiguration

Dieses Objekt wird in Kapitel 5 über den »MMPM/2« erläutert.

Win-OS/2-Konfiguration

Grundeinstellungen für die Win-OS/2-Umgebung werden hier vorgenommen. Näheres dazu im Kapitel 8.

MPTS-Netzwerkadapter und -Protokolldienste

Dieses Objekt verwaltet die Einstellungen der im System installierten Netzwerkadapter und deren Protokolle. Darauf wird im Kapitel 9 »Netzwerkfunktionalität« näher eingegangen.

Dienste für Benutzerprofilverwaltung

Über die Objekte dieses Ordners können Sie sich am Netzwerk anmelden und die Benutzer sowie deren Zugriffsrechte verwalten.

Systemstart

Die Objekte, die sich in diesem Ordner befinden, werden beim Systemstart automatisch gestartet. Der Unterschied zu einer Stapeldatei wie der »startup.cmd« liegt darin, daß die Startreihenfolge nicht festgelegt werden kann.

Möchten Sie den Start dieser Dateien unterbinden, fügen Sie einfach in der »config.sys« die Zeile »SET RestartObjects=NONE« ein.

Ein kleiner Tip: Erstellen Sie ein Programmobjekt für Ihren Lieblingseditor mit der »config.sys« als Startparameter. Eine Plazierung im Ordner *Systemkonfiguration* wäre geeignet.

Warp Center



Abb. 4.16: Das Warp Center

Das Warp Center ist die Funktionsleiste bzw. Statusleiste, die direkt nach der Installation am oberen Bildrand Ihrer Arbeitsoberfläche erscheint. Sie enthält Informationen über Ihr System und soll Ihnen über Fächer zu einem schnellen und übersichtlichen Zugriff auf häufig benutzte Objekte helfen. Der *OS/2 Warp*-Button am linken Rand zeigt eine Hierarchie der Objekte auf Ihren Arbeitsoberfläche an und wird natürlich automatisch aktualisiert. Daneben befinden sich Buttons für die Fensterliste, Sperren, Suchen und den Systemabschluß.

Der *Resource Monitor* informiert über Plattenbelegung, Systemlast und über den eventuellen Batteriestatus. Durch einen einfachen Klick kann zwischen den unterschiedlichen Ansichten gewechselt werden. Das Gleiche gilt für die am rechten Rand befindliche Uhr, Datumsanzeige und Stoppuhr. Die Mitte des Centers dient zur Ablage von Objekten, die hier per Drag&Drop eingebracht werden können. Durch einfachen Mausklick kann auch hier zwischen den verschiedenen Fächern (Tablett, Schublade) umgeschaltet werden. Das Icon neben der Uhr bietet Ihnen direkten Zugriff auf den Ordner *Unterstützung*.

Zur Anpassung des »Warp Centers« steht wie immer ein Einstellungs-Notizbuch zur Verfügung.

Unterstützung

Wie der Name schon sagt, finden Sie hier Unterstützung bei ihrer Arbeit mit OS/2. Diese wird durch das *OS/2 Warp Lernprogramm*, *Warp Guide*, den Ordner *Information* und via Internet bereitgestellt.

Insgesamt kann man von einer vielumfassenden Unterstützung reden. Sobald Sie nach der Installation das System starten, erhalten Sie die Möglichkeit, sich zu registrieren und das leistungsstarke »OS/2 Lernprogramm« auszuführen. Da Sie das wahrscheinlich nicht getan haben, bin ich in diesem Kapitel auch nur auf Grundlagen eingegangen.

Warp Guide

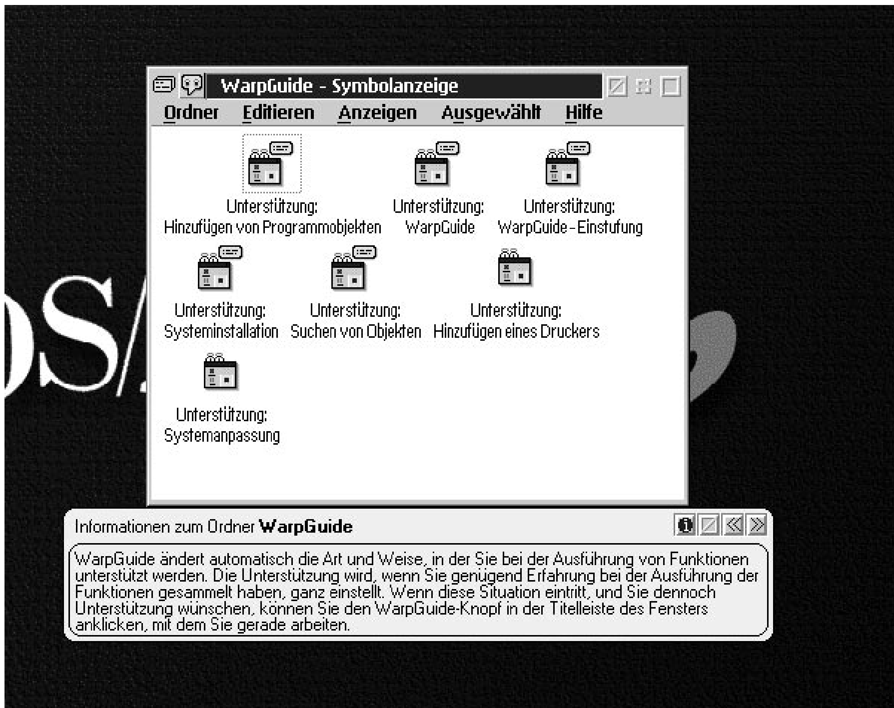


Abb. 4.17: Warp Guide in Aktion

Dem Warp Guide (frei übersetzt als »Warp-Führer«) sind Sie sicherlich nach ein paar Minuten mit OS/2 begegnet. Er soll sowohl dem OS/2-Warp-Version-4-Neuling als auch dem erfahrenen OS/2-Anwendern bei der Anpassungsarbeit mit hilfreichen Tips zur Verfügung stehen. Wird der Warp Guide zum ersten Mal gestartet, werden Sie zur Eingabe Ihres Namens und einer Einstufung Ihrer Computerkenntnisse gebeten. Abhängig von diesen wird Ihnen dann der »Warp Guide« in Zukunft zur Seite stehen. Über das Einstellungs-Notizbuch des *Warp Guide*-Ordners können Sie diesen konfigurieren und auch deaktivieren. Der Ordner selbst enthält

eine Reihe von Warp Guides für die wichtigsten Systemanpassungsfunktionen. So führt Sie zum Beispiel das Objekt *Unterstützung|Systemanpassung* durch eine Reihe von Einstellungs-Notizbüchern, ohne daß Sie extra danach suchen müssen.

Das Objekt *Softwareaktualisierungen* ruft den WebExplorer auf und erstellt eine Verbindung zum IBM-Internet-Server, von dem man – wie der Name schon sagt – Software-Updates beziehen kann. Zur »Fehlerbehebung« dienen die Objekte in diesem Ordner. Wird »AskPSP« installiert, ist es hier zu finden. Im Ordner *Information* finden Sie gut strukturiert mehrere Online-Handbücher. Bei den meisten handelt es sich um Dateien im IPF-Format, die mit Hilfe des Programmes »view.exe« angezeigt werden. Mit dem Programm »viewhelp.exe« können Sie auch »*.hlp«-Dateien betrachten.

4.7.2 Verbindungen

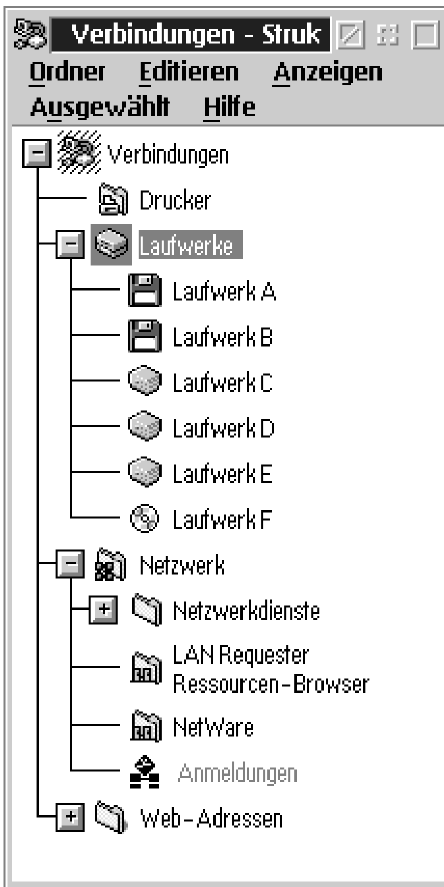


Abb. 4.18: Verbindungen

Um den Inhalt dieses Ordners zu erraten, muß man den Begriff »Verbindungen« sehr weitsichtig betrachten. Falls installiert, befinden sich in ihm die Netzwerkobjekte und auch standardmäßig die Drucker- und die Laufwerksobjekte.

Druckerobjekte

Druckerobjekte stellen die Druckeinheit samt ihrer Einstellungen dar. Dabei werden alle Druckaufträge, egal ob diese von einer OS/2-, DOS- oder Windows-Anwendung stammen, zentral vom OS/2-Drucksystem verwaltet.

Somit können Sie die Aktivitäten eines Druckers über sein Druckerobjekt kontrollieren. Für die Druckjobliste stehen Ihnen mehrere Anzeigemöglichkeiten zur Verfügung. Neben der Symbol- bzw. Detailanzeige eines Druckerobjekts können Sie über dessen Kontextmenü auch die *Anzeige aller Jobs* erhalten, die der Spooler momentan verwaltet.

Im Fach *Anzeige* des Einstellungs-Notizbuchs eines Druckerobjekts können Sie die Symbolbeziehungsweise Detailanzeige als Standardaktion beim Öffnen des Objekts markieren. Für die Detailanzeigen können Sie auch die anzuzeigenden Details wählen. Den Treiber für dieses Druckerobjekt wählen Sie im *Druckertreiber*-Fach aus. Dort können Sie auch die Drucker- und Jobmerkmale ändern. Jobmerkmale sind auch über den Kontextmenüpunkt *Öffnen* des Druckerobjekts erreichbar. Sie sollten beachten, daß es sich hierbei um unterschiedliche Einstellungsmöglichkeiten handelt, die abhängig vom Drucker(treiber) sind.

Im Fach *Ausgabeanschluß* können Sie die Schnittstelle angeben, an der der Drucker angeschlossen ist, sie konfigurieren sowie die Druckausgabe in eine Datei umleiten. Das Fach *Warteschlange* bietet Ihnen nun die Möglichkeit, die Druckpriorität für jedes Druckerobjekt explizit anzugeben.

Möchten Sie zwischen den verschiedenen Druckjobs eine Trennseite ausdrucken, können Sie diese im Feld *Trennseite|Datei* des Faches *Druckoptionen* angeben. Dort kann auch ein Zeitraum definiert werden, in dem Ausdrucke erfolgen. Das Registerfach *Textformat,* neu hinzugekommen in dieser OS/2-Version, dient dazu, reine Textdateien beim Ausdruck zu formatieren.

Bevor ein Ausdruck erfolgt ist, speichert der Spooler die formatierte Datei im Spoolerverzeichnis ab. Eines dieser temporären Datenddateiobjekte, die von Spooler erstellt werden, ist vom Typ »Metafile« und erscheint in der Druckjobliste. Über sein Kontextmenü können Sie den Druckjob löschen, kopieren und ihn über *Als nächstes Drucken* an die vorderste Stelle der Druckaufträge setzen. Mit einem Doppelklick kann der »Jobinhalt« geöffnet werden. Dabei wird das Programmobjekt *Bilddateien anzeigen* bzw. die Programmdatei »picview.exe« gestartet und der Jobinhalt angezeigt. Informationen über einen Druckjob erhalten Sie in seinen Einstellungen. Dort können Sie unter anderem seine Priorität, die Anzahl der Kopien und die Zeichenumsetzungstabelle ändern.

Tip: Möchten Sie ein neues Druckerobjekt installieren, können Sie dazu die Druckerschablone aus dem Schablonen-Ordner benutzen. Falls Ihnen die Druckertreiberdatei zur Verfügung steht, öffnen Sie sie und ziehen das geeignete Druckerobjekt auf die WPS, wodurch der Treiber eingebunden wird.

4.7.3 Programme

Hier finden Sie die vom System mitgelieferten Anwendungsprogramme. Diese sind in Ordnern sortiert.

Reißwolf

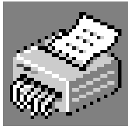



Abb. 4.19: Der Reißwolf

Der Reißwolf vernichtet alles, was Sie ihm übergeben.

4.8 Die Besonderheiten der Arbeitsoberfläche

Dieser »Ordner aller Ordner« stellt den Arbeitsbereich für den Benutzer dar. Als solcher besitzt er zusätzlich zu den Funktionen eines Ordners noch weitere, auf die ich hier näher eingehen werde.

Über sein Kontextmenü können Sie das System, um unbefugten Personen den Zugang zu verwehren, für Tastatur- und Mauseingaben »sperren«. Dabei wird auch die Tastenkombination [Strg]+[Alt]+[Entf] gesperrt. Wird diese Funktion zum ersten Mal aufgerufen, werden Sie zur Eingabe eines Kennworts aufgefordert. Dieses können Sie jederzeit in Fach *Sperren* des Einstellungs-Notizbuchs der Arbeitsoberfläche ändern. Dort können Sie auch den Sperrvorgang automatisieren sowie das Erscheinungsbild des gesperrten Systems beeinflussen. Seit OS/2 Warp Version 4 können Sie die Sperrfunktion auch offiziell als Bildschirmschoner einsetzen. Haben Sie das Kennwort vergessen, müssen Sie den Rechner neu starten und – sobald ein kleines Rechteck in der oberen linken Ecke des Bildschirms erscheint – die Tastenkombination [Alt]+[F1] drücken. Dadurch erhalten Sie den sogenannten »Wiederherstellungsbildschirm«, über den Sie mit [F2] zu einer OS/2-Befehlszeile gelangen. Dort geben Sie folgende Befehle ein:

– »cd os2«, um ins OS/2-Verzeichnis zu wechseln

– »makeini os2.ini lock.rc«, um das Kennwort auf den Standardwert zurückzusetzen.

Mit »exit« wird die Befehlszeile verlassen und der Rechner neu gebootet.

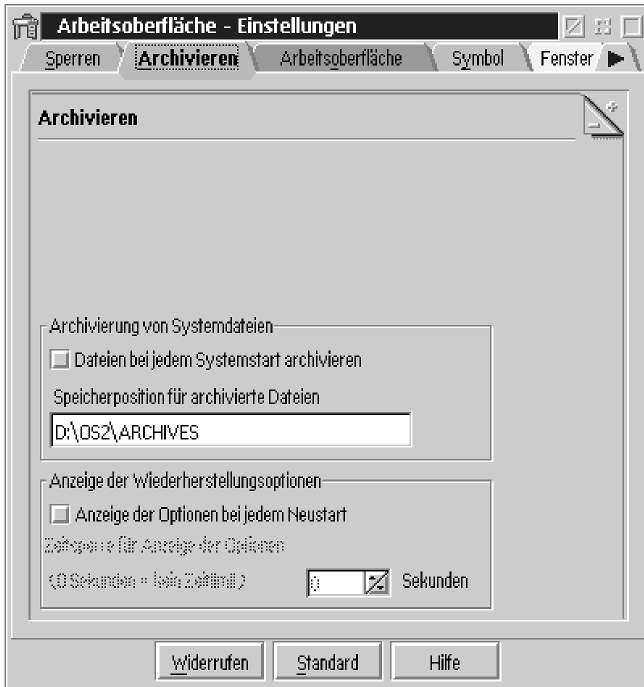


Abb. 4.20: Das Einstellungs-Notizbuch der Arbeitsoberfläche

Das Fach *Archivieren* dient zur Sicherung einer vorhandenen OS/2-Konfiguration. Diese wird mit dem Markierungsfeld *Dateien bei jedem Systemstart archivieren* aktiviert und findet im angegeben Verzeichnis beim nächstem Systemstart statt. Dadurch werden drei Generationen (01,02,03) gesichert (\OS2\ARCHIVES\0?). Falls schon drei Generationen vorhanden sind, wird bei der nächsten Sicherung die älteste überschrieben.Weil das Warp-Center aber nicht automatisch archiviert wird, muß die Datei »os2.key« im Verzeichnis »os2\archives\« editiert werden. Zu den Zeilen

```
KEYFILE:OS.INI
KEYFILE:OS2SYS.INI
KEYFILE:x:\CONFIG.SYS
KEYFILE:x:\STARTUP.CMD
KEYFILE:x:\AUTOEXEC.BAT
KEYFILE:x:\OS2INIT.CMD
```

fügen Sie noch folgende Zeilen ein:

```
Keyfile:c:\os2\dll\dock0.cfg
Keyfile:c:\os2\dll\dock1.cfg
Keyfile:c:\os2\dll\dock10.cfg
Keyfile:c:\os2\dll\dock11.cfg
Keyfile:c:\os2\dll\dock12.cfg
Keyfile:c:\os2\dll\dock13.cfg
```

```
Keyfile:c:\os2\dll\dock14.cfg
Keyfile:c:\os2\dll\dock15.cfg
Keyfile:c:\os2\dll\dock2.cfg
Keyfile:c:\os2\dll\dock3.cfg
Keyfile:c:\os2\dll\dock4.cfg
Keyfile:c:\os2\dll\dock5.cfg
Keyfile:c:\os2\dll\dock6.cfg
Keyfile:c:\os2\dll\dock7.cfg
Keyfile:c:\os2\dll\dock8.cfg
Keyfile:c:\os2\dll\dock9.cfg
Keyfile:c:\os2\dll\SCENTER.CFG
```

Mit der »os2.key« können Sie auch andere Dateien sichern lassen, beispielsweise die Konfigurations- und Ini-Dateien Ihrer wichtigsten Programme Ihres Netzwerks und natürlich auch Ihre Windows-Konfiguration.

Sie können die Datei also nach Belieben erweitern (vorher sollten Sie aber ihren Schreibschutz aufheben). Zum Wiederherstellen eines gesicherten Archivs wählen Sie dieses im *Wiederherstellungsbildschirm* an. Für das Aussehen des Wiederherstellungsbildschirms sind folgende Dateien aus dem »\os2\Boot\«-Verzeichnis zuständig:

»altf1top.scr«, »altf1mid.scr« und »altf1bot.scr«.

Was hinter der Fassade stecken soll, entscheiden die Datein »config.?« bzw. »altf1?.cmd« im gleichen Verzeichnis. Das Fragezeichen steht für den Buchstaben, mit dem diese Dateien aufgerufen werden. Wird die Rekonstruierung eines Archivs gewählt, sucht OS/2 im Verzeichnis, das in der Datei »archbase.$$$« angegeben ist, ob dieses existiert. Dort steht in der Datei »archives.$$$« die chronologische Reihenfolge der Archive. Sie können somit ein Archiv auf Ihrer Datenpartition sichern, um bei einer Neu-Installation des Systems den nötigen Konfigurationsaufwand zu verringern.

Wichtig ist, daß bei einer Archivierung zwar die OS/2-Konfigurationsdateien gesichert, aber nicht die Dateien, wie zum Beispiel Grafiktreiber etc., auf die sie zugreifen. Die Verzeichnisstruktur der WPS wird immer gesichert. Über das Feld *Einstellungen der Arbeitsoberfläche sichern* im Fach *Arbeitsoberfläche* können sie einstellen, ob die Änderungen, die Sie an Objekten der WPS vornehmen, gesichert werden. Ist das Feld aktiviert, werden beim Schließen eines Objektes dessen Einstellungen, die nicht in seinem Notizbuch vorgenommen werden, wie zum Beispiel die Fensterposition, gespeichert.

Unter *Standardübergabeoperation* können Sie angeben, was der Drag&Drop-Vorgang auf und von der Arbeitsoberfläche bewirken soll. Sie sollten auf der WPS keine Datendateien aufbewahren. Um auf häufig genutzte Objekte zuzugreifen, gibt es ja die Referenzen und Programmobjekte.

4.9 Tastaturabkürzungen

Tastatur	Maustaste	Aktion	Ort/Objekt	Wirkung
Strg	rechte	ziehen	Objekt	Kopieren
Strg	rechte	ziehen	Von Palette	Schriftfarbe ändern
Alt	rechte	ziehen	Von Palette	Systemweite Übergabe
Alt	rechte		Alle	Objektaufnahme
Shift	linke	Doppelklick	Alle	Objekt starten, wobei das Stammobjekt geschlossen wird
Strg+Enter	–	–	Objekt	s.o.
Strg+Shift	linke	Ziehen	Alle	Objekt-Referenz erstellen
Shift	linke	Doppelklick	Fensterliste	Verdecken/Minimieren
Alt	linke	Einfachklick	Alle	Objekttitel editieren
–	linke+rechte	Einfachklick	WPS	Fensterlistenaufruf
Strg	linke	Einfachklick	Alle	Mehrfachselektion
Strg	linke	Doppelklick	Fensterliste	Objekt maximiert in den Vordergrund holen
Entf	–	–	Fensterliste	Objekt schließen
Alt	linke	Doppelklick	Alle	Öffnen des Einstellungs-Notizbuchs
Strg	linke+rechte	Ziehen	Titelleiste	Fenster verschieben im Hintergrund
Alt+Shift+Tab	–	–	WPS	aktivieren
Shift+F9	–	–	Alle	Objekttitel editieren
Shift	rechte	Ziehen	Schablone	Verschieben
Alt+F1	–	–	Bootbeginn	Wiederherstellungsoptionen
Alt+F2	–	–	Bootbeginn	Ladevorgang der Treiber verfolgen

5 Der Multimedia Presentation Manager

von Norbert Heller

Im allgemeinen ist die Multimedia-Unterstützung besser in die WPS integriert als dies noch bei Warp V3 der Fall war.

Wav- (Midi-, AVI-, etc.) Dateien werden nun direkt abgespielt (ohne sichtbaren »Spieler«), wenn man auf die jeweilige Datei doppelklickt. Das Abspielen läßt sich beenden, indem man das Popup-Menü *Öffnen als|Stop* aufruft, oder den Ordner schließt, in dem sich die Datei befindet, die gerade gespielt wird.

Auch wurden die Multimedia-Applets optisch und funktionell »getunt«, was man spätestens beim CD- oder Wav-Audio-Player merkt.

5.1 Zu den Neuerungen im Einzelnen

– DART (Direct Audio RouTines)

Dies ist eine neue Methode zur Beschleunigung der Soundausgabe. Dies sollte für den »normalen Anwender« völlig transparent ablaufen (ein Sound klingt immer gleich, unabhängig davon, ob er mit der »alten« Methode der Playlisten gespielt wird, oder ob er über den DART-Treiber läuft). Jedoch ist ein DART-Treiber für die Soundausgabe wesentlich leichter zu programmieren als ein Playlisten-Treiber, was es dem Programmierer erleichtert, Spiele zu schreiben und bestehende Programme (z.B. von UNIX) auf OS/2 zu portieren.

– RTMIDI (Real Time MIDI)

Dies bedeutet eine bessere Kontrolle über die MIDI-Ausgabe, wie sie zum Beispiel für die Programmierung von MIDI-Sequenzern benötigt wird.

– Full Screen Dive

War die Video-Ausgabe bisher meist auf ein recht kleines Fenster auf der WPS beschränkt, so bietet sich nun die Möglichkeit, die Video-Ausgabe in einem Vollbild ablaufen zu lassen. Dies ist z.B. für Spiele interessant, da diese nun mit der gleichen Grafik-Performance wie unter DOS ablaufen können.

– Joystick-Unterstützung

Für Spiele ist die Unterstützung des Joysticks gegeben.

– 3D-Schnittstellen

Sie dienen als OpenGL-Unterstützung. Dies bedeutet hohe 3D-Ausgabegeschwindigkeit bei gleichzeitigem offenen, plattformunabhängigen Programmier-API, das eine schnelle Portierung von bestehender 3D-Software auf OS/2 ermöglicht.

– OPENDOC-Multimedia-Applets

MIDI- und Audio-/ Video-Player sind nun auch als OpenDoc-Parts verfügbar und können in OpenDoc-Applikationen eingebunden werden.

5.2 Allgemeine Anwendungen im MMPM/2-Paket

Lautstärkeregler

Mit dem Lautstärkeregler kann der Ton aus- oder eingeschaltet und die Lautstärke aller an das System angeschlossenen Multimediageräte gesteuert werden.

Bei vielen Multimedia-Programmen kann die Lautstärke auch separat geregelt werden. Dies ist zum Beispiel beim CD-Player oder dem Wav-Editor der Fall.

Multimedia-Vorlagen

Dies sind diverse Vorlagen, um beispielsweise eine neue Wav Datei zu erstellen, .

Da der Wave-Editor nicht mehr über den Menüeintrag *Datei|Neu* verfügt, muß eine neu zu erstellende WAV-Datei aus der Vorlagen-Schablone des Wav-Files erzeugt werden.

CD-Player

Mit dieser Multimedia-Anwendung können Audio-CDs abgespielt werden. Der CD-Player wurde optisch als auch funktionell überarbeitet, und besteht im Prinzip sogar aus zwei CD-Spielern.

Mit *Öffnen|Spieler* kommt man zu einen minimalen CD-Player, der die absoluten Basisfunktionen eines CD-Players bietet.

Mit *Öffnen|CD Spieler* kommt man zu einen wesentlich erweiterten CD-Player, der höheren Bedienungskomfort bietet. Im folgenden wird nur der erweiterte CD-Player besprochen, da sich die Play-Buttons des »einfachen« aus dem »erweiteren« ableiten lassen.

Play

Beim Drücken des Buttons *Play* wird das zur Zeit ausgewählte Lied abgespielt.

Stop

Beim Drücken des Buttons *Stop* wird das zur Zeit spielende Lied gestoppt.



Abb. 5.1:
Der erweiterte CD-Player

Pause

Bei Drücken von *Pause* wird das Abspielen des zur Zeit gespielten Lieds vorübergehend angehalten. Wird *Pause* erneut ausgewählt, wird der Abspielvorgang fortgesetzt.

Schneller Vorlauf / Rücklauf

Hier wird innerhalb des Lieds solange die Abspielposition vor- oder zurückgespult, wie der jeweilige Button gedrückt gehalten wird. War vorher *Play* aktiv, so wird das Spielen des Liedes an der neuen Position fortgesetzt.

Nächstes Lied

Bei dieser Auswahl wird das nächste Lied in der Abspielreihenfolge gespielt.

Vorheriges Lied

Bei Auswahl dieses Buttons wird das aktuelle Lied auf den Liedanfang zurückgesetzt

Eject

Durch Drücken des Buttons wird die CD-Schublade geöffnet (sofern dies vom CD-Laufwerk und vom Treiber unterstützt wird).

Track Buttons

Hier werden die Lieder angezeigt, die gespielt werden. Dies kann entweder die Original-Reihenfolge der CD sein, oder die Anzeige des aktuell spielenden Programms.

Status-Anzeige

Die Positions-Status-Anzeige besteht aus zwei Teilen :

Im linken Teil wird die Liednummer und die schon gespielte Zeit angezeigt. Im rechten Teil wird die Länge des ausgewählten Programms sowie die Gesamtzahl der Lieder angezeigt.

Schieberegler

Mit dem Schieberegler kann die Spielposition innerhalb der Lieder der CD (oder des aktuellen Programmes) dargestellt werden, oder es kann durch Betätigen des Schiebereglers auf eine andere Spielposition innerhalb der CD oder des Programms gesprungen werden.

Lautstärke

Durch Auswahl des Lautstärkereglers wird die Lautstärke des CD-Spielers gesteuert. Dies hat allerdings nur dann Effekt, wenn das CD-Laufwerk die Funktion des Lautstärkeregelns auch unterstützt, ansonsten wird der Ton an- oder ausgeschaltet. Lesen Sie hierzu mehr in der Beschreibung der Multimedia-Konfiguration Ihres CD-ROM-Laufwerks.

Menüoptionen

Titel Editieren

hier kann der aktuelle Titel der Audio-CD angegeben werden. Dieser Titel wird in der Titelleiste des Programms angezeigt und wird wiedererkannt, wenn die CD ein weiteres Mal eingelegt wird.

Der Name einer CD darf maximal 32 Zeichen lang sein.

Die Titel der CD und die Liedtitel werden in der Datei »cdp.ini« abgespeichert, die sich im Verzeichnis /MMOS2 befindet

Programm Editieren



Abb. 5.2: Titel der CD editieren

Hier können die Titel der einzelnen Lieder eingegeben werden. Dies erfolgt durch Betätigen von [Alt]+[linke Maustaste] auf dem Titeleintrag oder durch den Button *Editieren*.

Es kann ebenfalls ein »Programm« zusammengestellt werden, in welcher Reihenfolge welche Lieder gespielt werden sollen. Das Programm kann durch Drag&Drop der oberen Lieder in die untere Anzeige in der jeweiligen Reihenfolge zusammengestellt werden. Das gleiche gilt für den Button *Hinzufügen*.

Löschen

Das ausgewählte Lied in der unteren Anzeige wird aus dem Programm entfernt.

Freigeben

Löscht das ausgewählte Programm.

Standard

Es werden sämtliche Lieder auf einmal übernommen.

Umstellen

Hierbei werden die zum Programm zusammengestellten Lieder in willkürlicher Reihenfolge gemischt.

Sortieren

Hierbei werden die zum Programm zusammengestellten Lieder in aufsteigender Reihenfolge dargestellt

Der CD-Player besitzt mehrere Spielmodi:

Anspielen

Damit werden die Lieder der CD-ROM (des Programms) für fünf Sekunden angespielt, danach wird das nächste Lied gespielt.

Zufällige Abspielfolge

Die Lieder der CD-ROM (des Programms) werden in willkürlicher Abfolge gespielt.

Wenn ein Programm durch *Programm Editieren* zusammengestellt wurde, werden die Lieder in dieser Reihenfolge abgespielt.

Optionen des CD-Players:

Automatische Wiedergabe

Die CD wird abgespielt, wenn sie in das CD-Laufwerk eingelegt wird, ohne daß explizit *Play* gedrückt werden muß.

Position Wiederaufnehmen

Hier wird beim nächsten Start die CD bei dem Lied weitergespielt, an dem es gestoppt wurde.

Digitale Übertragung

Bei Auswahl der digitalen Übertragung werden die Musikdaten von der CD direkt an den Lautsprecherausgang der Soundkarte geleitet, ohne über die D/A zu gehen.

Diese Auswahl wird bei den Optionen nur dann angeboten, wenn das CD-ROM die Daten intern auch verarbeiten kann und eine Soundkarte angeschlossen ist. In allen anderen Fällen wird diese Auswahl ausgeblendet.

Die digitale Übertragung hat im allgemeinen eine höhere Ausgabequalität als die »normale« Datenübertragung.

5.3 Multimedia-Konfiguration zum CD-ROM-Laufwerk

In der Multimedia-Konfiguration zum CD-ROM-Laufwerk können einige Werte definiert werden, die bei Warp V3 noch nicht eingestellt werden konnten. Dies geschieht in der Notizbuch-Seite *Laufwerksfunktionen* zur CD-Konfiguration:

Möglichkeiten zum Laufwerk

Darin können einige Einstellungen zum CD-ROM-Laufwerk vorgenommen werden, wenn man der Meinung ist, daß die Standardeinstellungen, die nach dem »besten Schätzwert« vom System vorgenommen wurden, eventuell nicht die volle Leistungsfähigkeit des CD-Laufwerks ausschöpfen.

Diese jeweiligen Einstellungen funktionieren natürlich nur dann, wenn die CD-ROM-Hardware dies ebenfalls unterstützt, ansonsten tut sich bei dem jeweiligen Schalter eben nichts.

Ausgabe

Hier wird eingestellt, ob das CD-Laufwerk per Software-Kommando die Schublade ausfährt.

Manuelle Ausgabe inaktivieren

Hier wird eingestellt, ob das CD-Laufwerk per Software-Kommando den Eject-Knopf abschalten kann und somit die CD »gesperrt« ist .

Erst nach einem »entsperren« kann die CD-ROM über den Eject-Knopf wieder entnommen werden.

CD-Fach schließen

Damit wird eingestellt, daß das CD-ROM-Laufwerk per Software-Kommando die Schublade nicht nur öffnen, sondern auch schließen kann. Der CD-ROM-Treiber muß dies natürlich ebenfalls unterstützen.

Lautstärke-Regelung

Hier kann die Lautstärkenregelung des jeweiligen CD-Laufwerks besser angepaßt werden, so daß ein gleichmäßiger Lautstärkezuwachs über die Regelskala entsteht. Dies bedeutet: wenn der Lautstärkeregler auf 10% steht, soll die Lautstärke auch 10% von der Gesamtlautstärke betragen. Dies war bei der Warp V3 oft nicht der Fall, denn bei manchen CD-Laufwerken war

bei einer Regelung auf 10% schon die volle Lautstärke vorhanden.

Falls der Standardwert nicht dem tatsächlichen Wert entspricht, kann gewählt werden zwischen :

Ein / Aus	CD-ROM unterstützt nur Musik an/aus
Nur Ein	CD-ROM unterstützt nur volle Lautstärke
Systemvorgabe	CD-ROM wird auf den besten Schätzwert eingestellt
Niedrig	CD-ROM regelt die Lautstärke eher im unteren Skalenbereich
Proportional	CD-ROM regelt die Lautstärke gleichmäßig im Skalenbereich
Hoch	CD-ROM regelt die Lautstärke im oberen Skalenbereich
Sehr hoch	CD-ROM regelt die Lautstärke im äußersten oberen Skalenbereich

Um diese Einstellungen zu testen, kann mit dem CD-Player die Lautstärke beim Abspielen einer CD verändert werden.

Die Lautstärke sollte sich im besten Fall über die gesamte Lautstärkeskala regeln lassen.

Balance linker/rechter Kanal

Damit können die Lautstärken der beiden Kanäle separat geregelt werden. Ist dies nicht der Fall, wird die lauteste Einstellung der beiden Kanäle zur Lautstärkeeinstellung beider Kanäle verwendet.

Digitale Übertragung

Bei dieser Auswahl werden die Musikdaten von der CD direkt an den Lautsprecherausgang der Soundkarte geführt, ohne daß über die D/A-Wandler gegangen wird.

Die Auswahl *Digitale Übertragung* wird bei den Optionen nur dann angeboten, wenn das CD-ROM die Daten intern auch verarbeiten kann und eine Soundkarte angeschlossen ist. In allen anderen Fällen wird die Auswahl der digitalen Übertragung ausgeblendet.

Suchen

hier kann eingestellt werden, ob das CD-ROM an eine bestimmte Stelle »fahren« kann, die per Software adressiert wird.

Die meisten CD-ROMs besitzen zwar diese Fähigkeit, es kann jedoch zu Performance-Verbesserungen führen, wenn dieses Kommando angeschaltet wird.

UPC

Der UPC (Universal Product Code) ist ein bestimmter Wert, der sich auf jeder CD-ROM findet. Manche Software benutzt nur den UPC und kann etwas damit anfangen (z.B. Lieder aus einer CD-Datenbank holen).

Manche CD-ROM-Laufwerke können den UPC auslesen, und tun dies auch richtig. Andere CD-ROM-Laufwerke meinen zwar, sie könnten, liefern aber völlig falsche Werte. Hier sollte die UPC-Erkennung abgeschaltet werden, ebenso wie bei CD-ROM-Laufwerken, die den UPC gar nicht lesen können.

5.4 Der Bildbetrachter

Mit dem Bildbetrachter kann ein Bild in diversen Formaten direkt auf der WPS betrachtet und in ein anderes Format konvertiert werden.

Darin finden sich auch einfache Bildmanipulationsmöglichkeiten:

Mit »gedrückthalten und ziehen« der linken Maustaste läßt sich eine Selektion erstellen. Diese Selektion kann man per Drag&Drop in ein anderes, geöffnetes Bild ziehen, speichern oder bearbeiten.

Bild Sichern unter

Damit wird das aktuelle Bild (und gegebenfalls Änderungen) gespeichert.

Einfügen

Fügt einen Bildbereich aus dem Clipboard in den selektierten Bereich ein.

Kopieren

Hier kann ein Bildbereich ins Clipboard kopiert werden.

Ausschnitt erstellen

Damit wird das Gesamtbild auf den Selektionsausschnitt reduziert.

Statuszeile

Sie stellt beispielsweise dar, aus wievielen Farben sich dieses Bild zusammensetzt, und ob das Bild auch mit mehr oder wenigeren Farben dargestellt werden kann.

Mit den Pfeilen »<<« und »>>« kann das gleiche Bild in 16 oder 256 Farben dargestellt werden, oder – (falls vorhanden – , aus mehreren Bildseiten eine ausgewählt werden.

An Fenster anpassen

Das Bild wird an den Darstellungsbereich des Fensters angepaßt, also auf die Größe des Fensters skaliert.

5.5 Digitaler Audio-Spieler

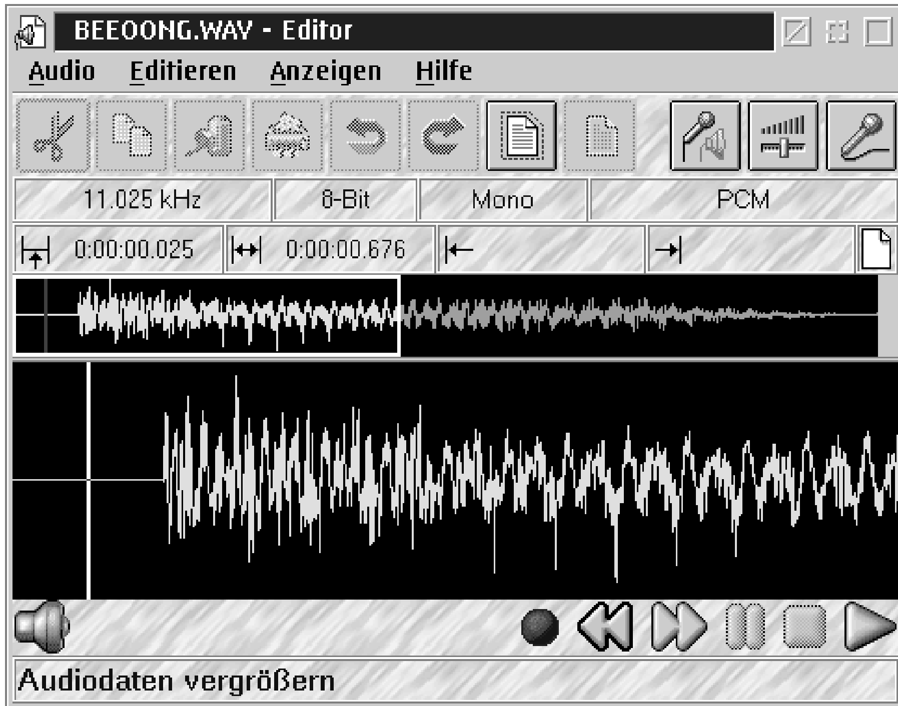


Abb. 5.3: Der Wave-Editor

Mit dieser Anwendung können Wav-Dateien gespielt oder es können verschiedene Editierfunktionen auf die Wav-Datei angewendet werden.

Start-Button

Die Datei wird ab der aktuellen Position abgespielt.

Stop-Button

Das Spielen der Datei wird gestoppt.

Pause-Button

Die Wiedergabe bzw. Aufnahme wird vorübergehend gestoppt.

Um die Wiedergabe bzw. Aufnahme wiederaufzunehmen, muß erneut *Pause* oder *Start* bzw. *Aufnahme* ausgewählt werden.

Aufnahme-Button

Damit wird ein neues Lied aufgenommen.

Rücklauf-Button

Die Abspielposition wird auf den Anfang des Liedes gesetzt

Vorlauf-Button

Die Abspielposition wird auf das Ende des Liedes gesetzt

Menüpunkte

Audio Sichern

Die Wav-Datei wird unter ihrem aktuellen Dateinamen gespeichert.

Audio sichern Unter

Die Datei wird mit einem neuen Namen gesichert.

Mikrofon konfigurieren

Wenn über ein Mikrofon aufgenommen werden soll, ist *Mikrofon* im Menü *Optionen* auszuwählen. Es wird dann der Eingangspegel des Mikrofons bei den Optionen eingestellt. Über *Ton überwachen* wird die aktuelle Aufnahmelautstärke (z.B. über Kopfhörer) überprüft.

Für die Eingabeeinheit kann ein niedriger, mittlerer oder hoher Eingangspegel ausgewählt werden. Je höher der Pegel ist, desto empfindlicher ist das Mikrofon.

Line-In konfigurieren

Über den Line-In-Eingang kann ein Signal eines CD-ROMs oder eines Tonbandes aufgenommen werden (Menüpunkt *Eingang*.)

Bei den Optionen wird der Eingangspegel des Line-In-Eingangs eingestellt. Über *Ton überwachen* wird die aktuelle Aufnahmelautstärke überprüft.

Aufnahmequalität und Speicherformate

Stereo/Mono

Die Aufnahme wird in Stereo- bzw. Monoqualität aufgenommen.

Niedrige Qualität

Für die Aufnahme wird eine Sampling-Rate von 8000 Hz verwendet. Dies entspricht der niedrigsten Qualitätsstufe.

Sprache

Für die Aufnahme wird eine Sampling-Rate von 11025 Hz verwendet. Dies entspricht einer niedrigen Qualitätsstufe.

Musik

Für die Aufnahme wird eine Sampling-Rate von 22050 Hz verwendet. Dies entspricht einer mittleren Qualitätsstufe.

HiFi

Für die Aufnahme wird eine Sampling-Rate von 44100 Hz verwendet. Dies entspricht einer hohen Qualitätsstufe.

16-Bit dekomprimiert

Die Audiodaten werden mit einer Sample-Rate von 16 Bit unkomprimiert abgelegt.

8-Bit dekomprimiert

Die Audiodaten werden mit der Sample Rate von 8 Bit unkomprimiert abgelegt.

4-Bit IMA-ADPCM-Komprimierung

Die Audiodaten werden mit der Sample Rate von 4 Bit mit Datenkompression abgelegt.

Bereich markieren

Für einige der Editierbefehle im Menü *Editieren* muß ein Bereich innerhalb der aktuellen Datei markiert oder ausgewählt werden. Es ist möglich, die gesamte Datei oder nur einen bestimmten Bereich innerhalb der Datei zu markieren. Um die gesamte Datei zu markieren, muß *Alles auswählen* im Menü *Editieren* ausgewählt werden.

Nachdem ein bestimmter Bereich der Datei markiert wurde, können verschiedene Effekt auf diesen Bereich angewendet werden:

ECHO:

Hier wird ein Echo auf den markierten Bereich angewendet. Es kann die Echo-Verzögerung (0-100ms) eingestellt werden, sowie die Lautstärke des Echo(0 bis 100%).

Nachhall

Hier wird ein Nachhall auf den markierten Bereich angewendet. Es kann die Nachhall-Verzögerung (0-100ms) eingestellt werden, sowie die Lautstärke des Nachhalls (0 bis 100%).

Einblenden

Die Lautstärke wird übergangsweise von Null auf das Maximum gesetzt, so daß langsam in diesen Bereich eingeblendet wird.

Ausblenden

Die Lautstärke wird übergangsweise von Maximum auf Null gesetzt, so daß langsam aus diesem Bereich ausgeblendet wird.

Rückwärts

Der markierte Bereich wird »umgedreht«, also rückwärts gespielt.

Doppelte Geschwindigkeit

Der markierte Bereich wird mit doppelter Geschwindigkeit abgespielt. Dies erhöht außerdem die Tonhöhe des Bereiches.

Halbe Geschwindigkeit

Der markierte Bereich wird mit der halben Geschwindigkeit abgespielt. Dies erniedrigt die Tonhöhe des Bereiches.

Maximale Lautstärke

Der markierte Bereich wird auf die maximale Lautstärke gebracht, die ohne Clipping (Übersteuerung) möglich ist.

Lautstärke um 25 Prozent erhöhen/erniedrigen

Im markierten Bereich wird die Lautstärke um 25 Prozent erhöht/erniedrigt.

Mischen (mit Datei)

In den markierten Bereich wird eine Datei neu mit eingemischt.

Mischen (mit Zwischenablage)

In den markierten Bereich wird ein zuvor in die Zwischenablage kopierter Bereich eingemischt.

5.6 Midi-Spieler

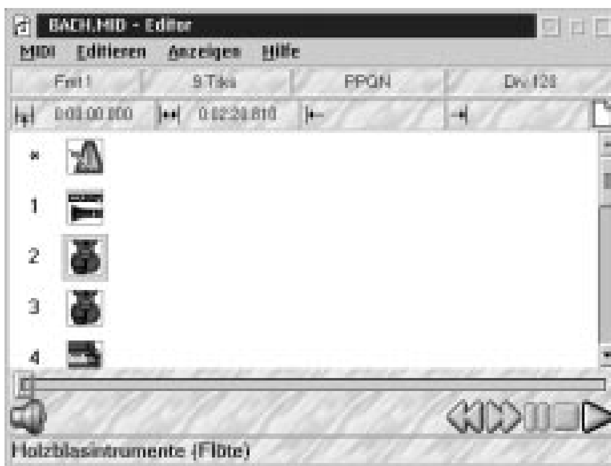


Abb. 5.4: Der Midi-Player

Start-Button

Die Datei wird ab der aktuellen Position abgespielt.

Stop-Button

Das Spielen der Datei wird gestoppt.

Pause-Button

Die Wiedergabe wird vorübergehend gestoppt.

Rücklauf Button

Die Abspielposition wird auf den Anfang des Liedes gesetzt

Vorlauf Button

Die Abspielposition wird auf das Ende des Liedes gesetzt

Menüoptionen

Midi Sichern

Hier wird die Datei unter ihrem aktuellen Dateinamen gespeichert.

Midi Sichern Unter

Die Datei wird mit einem neuen Namen gesichert werden.

Editor-Anzeige

Hier kann die Darstellung vom normalen Midi-Abspieler zu einem Instrumenten-Editor gewechselt werden.

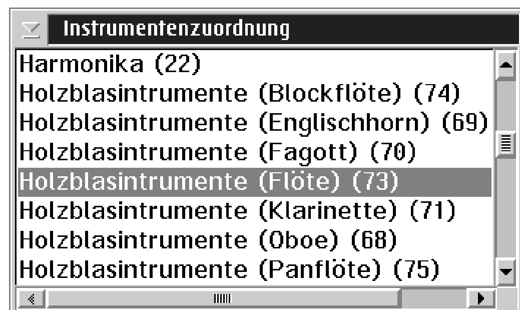


Abb. 5.5: Der Instrumenten-Editor

Durch Doppelklick mit der linken Maustaste auf ein Instrument kann dieser Midi-Spur ein neues Instrument zugeordnet werden.

Die einzelnen Noten der jeweiligen Instrumente können hiermit allerdings nicht geändert werden.

Tempoänderung des Midi-Files

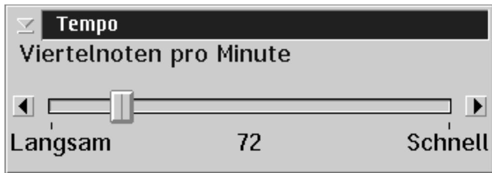


Abb. 5.6: Abspieltempo der Midi-Dateien variieren

Durch Doppelklick mit der linken Maustaste auf das Geschwindigkeits-Symbol kann das Abspieltempo der Midi-Datei verringert oder erhöht werden. Bei einer Tempoänderung bleiben die Tonhöhe und die Qualität der Musik erhalten.

Midikanäle verschieben



Abb. 5.7: Midikanäle verschieben

Ein Midi-File kann bis zu 16 Kanäle enthalten, jeder davon kann ein unterschiedliches Instrument enthalten.

Durch Doppelklick mit der linken Maustaste auf die Zahl vor dem angezeigten Instrument kann mit einem Dialog das jeweilige Instrument auf einen neuen Kanal verschoben werden.

Je niedriger die Kanalnummer ist, desto geringer ist dessen Priorität.

5.7 Rexx und Multimedia

Rexx kann alles, auch Multimedia.

Für Multimedia mit Rexx ist die normale Rexx-Installation natürlich Voraussetzung.

Hier ein paar Beispiele, um mit Rexx Multimedia-Dateien abzuspielen.

5.7.1 MIDI-Files abspielen

```
/*
 midi.cmd
*/

/*
    die benötigten REXX-Funktionen laden
*/
rc = call RxFuncAdd('mciRxInit', 'MCIAPI', 'mciRxInit')
rc = call mciRxInit()

/*
    den Dateinamen holen
*/
parse arg midi

if midi = '' then do
    say 'Aufruf: midi name.mid'
    exit 0
end

/*
    Das Device öffnen
*/
rc = mciRxSendString('open sequencer alias midi2 wait', 'rs', '0', '0')
if rc  0 then do
    rc2 = mciRxGetErrorString(rc, 'ErrStVar')
    say 'ERROR! rc =' rc ', ErrStVar =' ErrStVar
    exit rc
    end

/*
    Die Datei laden
*/
rc = mciRxSendString('load midi2 ' midi ' wait','rs', '0', '0')

Say "Spiele : " midi

/*
    Die Datei spielen
*/
rc = mciRxSendString('play midi2 wait','rs', '0', '0')

/*
    Wieder schliessen
*/
rc = mciRxSendString('close midi2 wait','rs', '0', '0')

call mciRxExit
exit
```

5.7.2 Wav-Files abspielen

```
/*
 wave.cmd
*/

/*
    die benötigten Rexx-Funktionen laden
*/
rc = call RxFuncAdd('mciRxInit', 'MCIAPI', 'mciRxInit')
rc = call mciRxInit()

/*
    den Dateinamen holen
*/
parse arg wave

if wave = '' then do
    say 'Aufruf: wave name.wav'
    exit 0
end

/*
    Das Device öffnen
*/
rc = mciRxSendString('open waveaudio alias wave wait', 'rs', '0', '0')
if rc  0 then do
    rc2 = mciRxGetErrorString(rc, 'ErrStVar')
    say 'ERROR! rc =' rc ', ErrStVar =' ErrStVar
    exit rc
    end

/*
    Die Datei laden
*/
rc = mciRxSendString('load wave ' wave ' wait','rs', '0', '0')

Say "Spiele : " wave

/*
    Die Datei spielen
*/
rc = mciRxSendString('play wave wait','rs', '0', '0')

/*
    Wieder schliessen
*/
rc = mciRxSendString('close wave wait','rs', '0', '0')

call mciRxExit
exit
```

5.7.3 Video-Files abspielen

```
/*
 video.cmd
*/

/*
    die benötigten REXX Funktionen laden
*/
rc = call RxFuncAdd('mciRxInit', 'MCIAPI', 'mciRxInit')
rc = call mciRxInit()

/*
    den Dateinamen holen
*/
parse arg avi

if avi = '' then do
    say 'Aufruf: pmrexx video name.avi'
    exit 0
end

/*
    Das Device öffnen
*/
rc = mciRxSendString('open digitalvideo alias video wait', 'rs', '0', '0')
if rc  0 then do
    rc2 = mciRxGetErrorString(rc, 'ErrStVar')
    say 'ERROR! rc =' rc ', ErrStVar =' ErrStVar
    exit rc
    end

/*
    Die Datei laden
*/
rc = mciRxSendString('load video' avi, 'rs', '0', '0')

/*
    Das Geschwindigkeitsformat setzen
*/
rc = mciRxSendString('set video speed format percentage wait','rs', '0', '0')

/*
    Die Datei spielen mit normaler Geschwindigkeit
*/
rc = mciRxSendString('play video speed 100 wait','rs', '0', '0')

/*
    Wieder schliessen
*/
rc = mciRxSendString('close video wait','rs', '0', '0')

call mciRxExit
exit
```

5.7.4 CD abspielen

```
/*
 playcd.cmd
*/


/*
    die benötigten Rexx-Funktionen laden
*/
call RxFuncAdd 'mciRxInit', 'MCIAPI', 'mciRxInit'
call mciRxInit

/*
   Das CD-ROM-Laufwerk öffnen
   Falls das CD-ROM das zweite CD-ROM ist, kann
    CDAudio02 angegeben werden

*/
rc = mciRxSendString('Open CDAudio alias CD wait', 'rs', '0', '0')
if rc  0 then do
        rc2 = mciRxGetErrorString(rc, 'ErrStVar')
        say 'ERROR! rc =' rc ', ErrStVar =' ErrStVar
        exit rc
    end

/*
    das Zeitformat setzen
*/
rc = mciRxSendString('Set CD Time Format Milliseconds wait', 'rs', '0', '0')

say "Spiele CD ab"

/*
    anfangen zu spielen
*/
rc = mciRxSendString('play CD wait', 'rs', '0', '0')

/*
    wieder schliessen
*/
rc = mciRxSendString('close CD wait','rs', '0', '0')

call mciRxExit
exit
```

5.8 Zusätzliche Programme und Codecs rund um OS/2-Multimedia

CODECS

Um eine Multimedia-Datei in einem bestimmten Format für das MMPM/2-System zur Verfügung zu stellen, wird ein sogenannter CODEC (COmpressor/DECompressor) eingesetzt. Dieser fungiert unter OS/2 als Schnittstelle zwischen dem Multimedia-System und der »eigentlichen« Datei.

Wird ein CODEC im MMPM/2 registriert, so können alle Multimedia-Programme diesen CODEC nutzen (den neuen Dateityp abspielen).

Im Gegensatz hierzu gibt es unter OS/2 auch noch die Möglichkeit, Sound- und Video-Formate in einem speziellen Programm abzuspielen (zum Beispiel manche MPEG-Player), ohne daß hierzu ein CODEC für das System registriert wird.

5.8.1 Windows-AVI-Files abspielen

Andreas Porteles AnPoCODEC ist eine Zusammenstellung von Video- (AVI) CODECs für den MMPM/2. Es werden momentan fünf Formate unterstützt: »AnPoMSVC« unterstützt das Video1 8-Bit- und 16-Bit-Format (VfW 1.0), »AnPoCVID« das Cinepak- (VfW 1.1) und Cinepak-8-Bit- (Win95) Format, »AnPoRLE« (VfW 1.0) das RLE-Format. Hiermit sollte es möglich sein, die »meisten« der zur Zeit gängigen Windows-AVI-Files abzuspielen.

Status: Shareware

Openmpeg (MPEG)

Von IBM kommt ein CODEC, der es erlaubt, VIDEO-MPEG-Files im Format MPEG 1 abzuspielen.

Status: Generell frei verfügbar (Mailboxen und Internet)

Apples Quicktime

Von der Firma Practice Corp. kommt ein Software-Codec, der es erlaubt, Apple-Quicktime-Movies abzuspielen.

Hierbei werden die Formate QuickMotion Animation, Cinema, Graphic, Photo, Snapshot, Video, Indeo21, Indeo32 und RGB unterstützt, und selbstverständlich auch die EnDive-Funktionen der jeweiligen Grafikkarten.

Status: Kommerziell, ca 80,– DM

5.8.2 Abspiel-Programme

AUDIO MPEG

Hier existiert ein Programm namens »Maplay«, mit dem es möglich ist, MPEG-Audio-Dateien unter OS/2 abzuspielen.

Hier kann wahlweise die Soundausgabe über den DART-Treiber erfolgen. Ebenfalls im Paket befindet sich ein Konverter, um diese MPEG-Audio-Dateien zu erzeugen. Beispielsweise kann eine WAV-Datei, die von CD gesampelt worden ist, in das Audio-MPEG-Format konvertiert werden. Der Kompressionsfaktor beträgt hier ca. 10:1, wobei keinerlei Unterschiede zum Original-CD-Klang festzustellen sind.

Um das Audio-MPEG abzuspielen, bedarf es allerdings eines schnellen Prozessors (Pentium 100 oder höher), sowie einer Soundkarte, die 44kHz und 16 Bit in Stereo abspielen kann (eben CD-Qualität).

Status: Freeware

5.8.3 Wave-Editoren

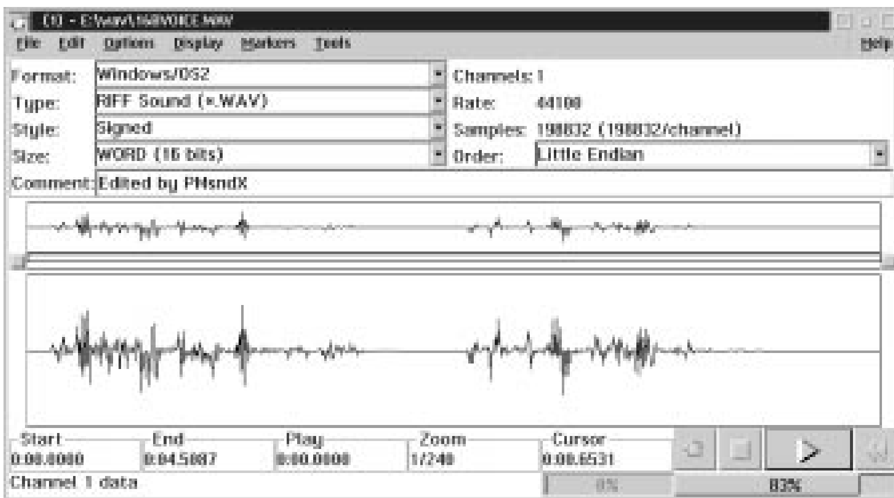


Abb. 5.8: Der SoundExchanger

PMSoundExchanger

Von Wishware gibt es den »PMSoundExchanger«. Dieser Sound-Editor bietet einige Transformationseffekte mehr als der IBM-Wav-Editor. Mit PMSoundExchanger können außerdem Sound-Files in verschiedenen Formaten unter OS/2 abgespielt werden.

Status: Shareware (Autor teilt allerdings auf Anfrage den Registrier-Schlüssel kostenlos mit)

5.8.4 AMIGA-MOD-Files (MOD/S3M/MTM/XM/ULT)

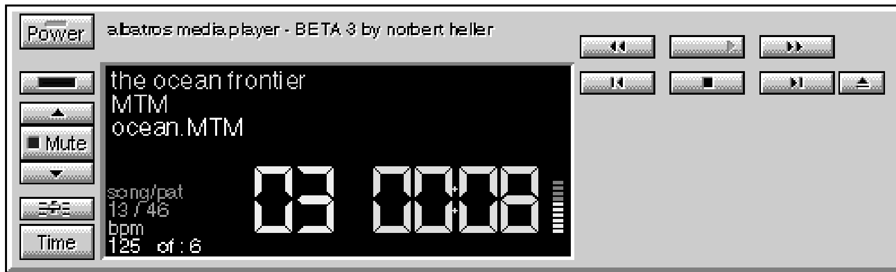


Abb. 5.9: Albatros Media Player

Albatros Media Player

Der Albatros Media Player kann dazu benutzt werden, diese Amiga-Modul-Formate abzuspielen.

Hierbei kann die Soundausgabe wahlweise über den DART-Treiber oder den normalen MMPM/2-Treiber angewählt werden.

Weiterhin können mit diesem Media-Player sämtliche OS/2-MMPM-Formate abgespielt werden, auch diejenigen Formate, die erst mit zusätzlichen CODECS wie z.B. Apcodec, Quickmotion oder OpenMpeg installiert werden.

Status: Shareware

5.8.5 MOD-File-Editor

PMPOZE

Das Programm PMPOZE kann dazu verwendet werden, MOD-Files unter OS/2 zu erstellen. Hier können auch aus bestehenden MOD-Files die Samples »gerippt«, und dann in eigenen Projekten weiterverwendet werden.

Status: Shareware

5.8.6 MIDI-Sequenzer

OS/2 Midi Sequencer

Far Pavillions Studio und Jim Bell stellen den OS/2 Midi Sequencer her. Dieser erfordert jedoch den neuen »HI-RES«-Timer-Treiber (der dem Paket beiliegt), sowie einen externen Synthesizer (liegt nicht bei).

Status: Shareware

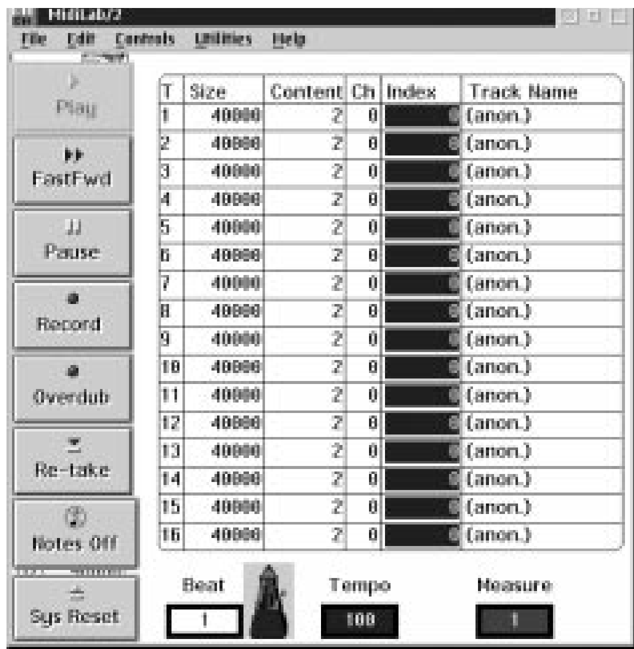


Abb. 5.10:
Der Midi Sequencer

5.8.7 MIDI-Composer

Nota Musica

Nota Musica ist ein Midi-Composer, der bisher noch nicht mit einem externen Synthesizer zusammenarbeitet. Hiermit können allerdings die Musiknoten auf Papier gebracht werden (MusicTex), wobei die Noten in einem Grafikmodus gesetzt werden können. Es wird die komplette Notation unterstützt.

Status: Shareware

5.8.8 CD-Player

Albatros CD Player

Albatros CD Player ist ein CD-Player, der wesentlich mehr Funktionen bietet als der bei OS/2 mitgelieferte CD-Player:

– Repeat Track

– Repeat Disk

– Random Play

– Intro Play

– Program Play

– A-B-Play

– Titeldirektauswahl

– speichert für jede CD das Programm ab

– LCD-Anzeige (konfigurierbar)

– Erkennt CD-Titel und Track-Titel wieder

– Zeigt CD Titel und Track Titel während der Spielzeit an

– Zeigt Track und Lietitel als Icon an

– Deutsche und englische Version

– Titelprogrammierung per Drag&Drop

– Farbselektionen per Drag&Drop

– Zeigt Informationen zur CD an

– Viele Anzeigeoptionen (Programmspielzeit, Restspielanzeige)

– ca. 400 fertig eingegebene CD-Titel und Liedtitel (Shareware-Version)

Status: Shareware



Abb. 5.11: Albatros CD-Player

5.8.9 CD- nach Wav-Konverter

PMCD2Wav

Mit dem Programm PMCD2Wav ist es möglich, Audio-CD-Tracks unter OS/2 als Wav-Datei abzuspeichern. Die Wav-Datei wird im Format 44 kHz / 16-Bit-Stereo abgespeichert. Das CD-Laufwerk, das zum Lesen benutzt wird, muß das »digitale Auslesen« unterstützen. Leider sind sehr viele billige CD-Laufwerke hierzu nicht geeignet.

Status: Shareware

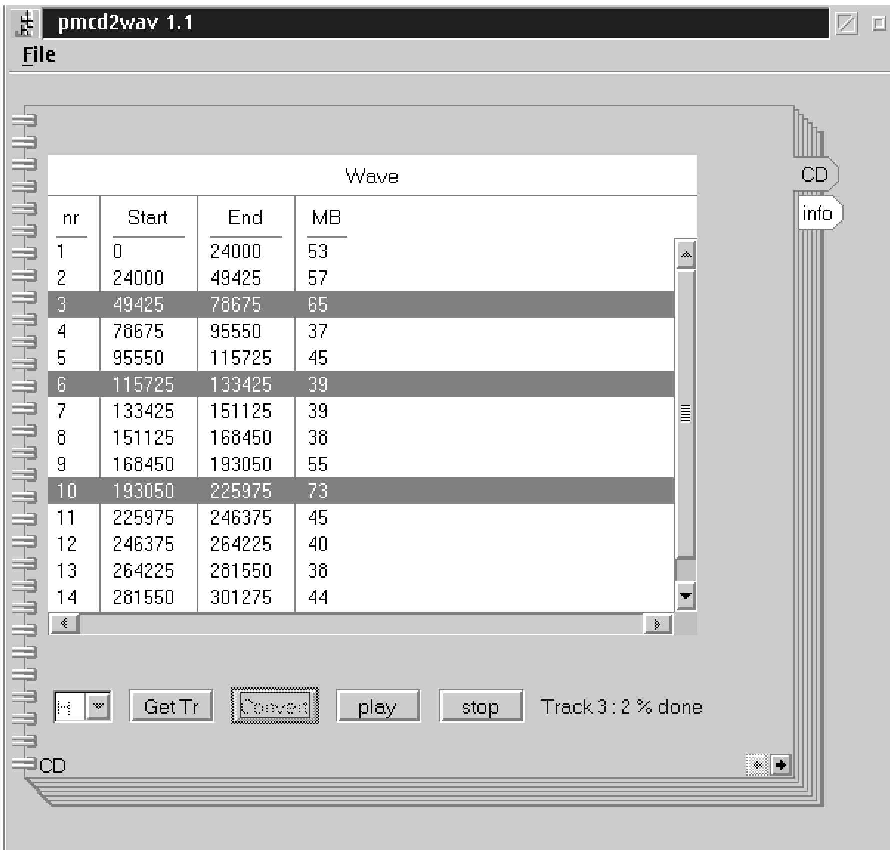


Abb. 5.12: PMCD2Wav

5.9 Soundkarten unter OS/2

5.9.1 Grundlegende Technologie

Soundkarten gibt es heutzutage in zwei Bauweisen: FM Synthese und Wavetable.

Eine FM-Synthese-Karte ist beispielsweise die klassische SoundBlaster-8-Bit- und 16-Bit-Karte. Diese kann digital aufnehmen, Soundfiles wiedergeben, MIDI-Dateien abspielen und in etwa auch Musikinstrumente durch Zusammenfügen verschiedener Sinuswellen emulieren.

Wavetable-Karten können auch immer digitalen Sound abspielen, zusätzlich jedoch können sie MIDI-Files per »Wavetable« abspielen. Hierbei werden also digitale Samples der jeweiligen Musikinstrumente dazu benutzt, das jeweilige Musikstück noch besser klingen zu lassen. Die jeweiligen Instrumenten-Samples befinden sich im ROM (oder RAM) der Soundkarte und können zum Teil auch von Festplatte in das RAM der Karte nachgeladen werden.

Aus diesem Grund klingt eine Wavetable-Karte wesentlich besser, wenn es darum geht, MIDI-Files zu erstellen oder abzuspielen, oder auch die Soundeffekte von Computerspielen wiederzugeben.

Keine Vorteile jedoch hat eine Wavetable-Karte, wenn man nur digitale Sounddaten wiedergeben möchte, z.B. eine WAV-Datei abspielt oder den Sound eines MPEG-Videos anhören möchte.

Wavetable-Karten können sich zum Teil sehr in der jeweiligen Qualität ihrer gespeicherten Musikinstrumente unterscheiden.

Das jeweilige Feature einer Soundkarte erfordert unter OS/2 Treiber. So gibt es immer noch viele Wavetable-Karten, deren OS/2-Treiber nicht das »Wavetable«-Feature unterstützt, somit klingt Midi-Musik mit dieser Karten- und Treiberkombination nicht besser als mit einer FM-Karte.

Auch unterstützen viele OS/2-Treiber nur begrenzt die Soundunterstützung der DOS- und Windows-Programme, die unter OS/2 laufen. Hier kann es oft zu Problemen mit Spielen kommen, die keine Soundausgabe in der DOS-Box erhalten.

5.9.2 MIDI

Viele (nicht alle) Wavetable-Karten benutzen das sogenannte MPU-401-Interface, um Midi-Dateien abzuspielen. Manche bevorzugen eine Soundkarte ohne MPU-401-Interface, da diese Methode die Anzahl externer Midi-Geräte, die an die Karte angeschlossen werden können, um eines erniedrigt.

IBM hat nun einen generellen MPU-401-Treiber, der theoretisch sofort alle Soundkarten mit diesem Interface unterstützen sollte.

Falls es nicht gelingen sollte, den IBM-MPU-401-Treiber zu installieren, obwohl dieses Interface auf der Karte vorhanden ist, sollte versucht werden, erst einmal nur die normale WAV-Unterstützung der jeweiligen Karte zu installieren, ohne die dazugehörige MIDI-Unterstützung.

Es kann auch versucht werden, den MPU-401-Treiber vor dem WAV-Treiber zu installieren.

Weiterhin kann es möglich sein, daß der MPU-401 erst per Software initialisiert wird. Dann bleibt meist nichts anderes übrig, als erst normales DOS zu starten, den MPU-401 zu initialisieren, und danach (ohne Betätigen des Reset-Knopfs) OS/2 zu starten.

Der MPU-401-Treiber läuft nur unter Warp V3 oder höher, nicht jedoch unter den Versionen OS/2 2.1 oder niedriger.

5.9.3 Chipsätze

Es scheint, als würden nur wenige Firmen wie OPTi oder Crystal Semiconductor die Chipsätze herstellen, die in den meisten Soundkarten zu finden sind. Beispielsweise stellen die OPTi-928- und 929-Chipsätze die Schnittstelle zu den eigentlichen »soundgenerierenden« Chips von Yamaha, Crystal Semiconductor und anderen dar.

Deshalb funktionieren Treiber der einen Firma oft auch mit einer Soundkarte einer anderen Firma.

5.9.4 DSP

Einige Soundkarten besitzen einen sogenannten DSP (Digitaler Signal-Prozessor), bei Creative Labs auch »ASP« genannt. DSPs sind Chips, die in einer Vielzahl von Anwendungen eingesetzt werden können.

So finden sich DSPs zum Beispiel in Modems, in Karten zur Signalerfassung und vielen anderen Dingen.

IBMs MWAVE-Chip ist ein DSP, der sich auf einer Karte befindet, die sowohl als Soundkarte als auch als Modem benutzt werden kann.

Die meisten dieser Multifunktionskarten haben jedoch leider keinen OS/2-Treiber (abgesehen von der IBM-MWAVE-Karte), was ihren Nutzen für OS/2 natürlich einschränkt.

5.9.5 Spracherkennung

Die Spracherkennung war schon mehrere Jahre für OS/2 verfügbar, allerdings dann nur in Verbindung mit einer speziellen Karte.

Inzwischen jedoch wurden die Hardware-Voraussetzungen für die Spracherkennung so weit heruntergedrückt, daß diese mit einem Pentium und einer »normalen« Soundkarte funktioniert.

Laut IBM soll diese Spracherkennung auch auf einem 486er Prozessor möglich sein, wenn eine IBM-Mwave-Karte eingesetzt wird.

5.9.6 Multi-Instance-Unterstützung

Viele Soundkarten erlauben es nicht, mehr als einen »Sound« gleichzeitig zu spielen. Bei einem Multitasking-System wie OS/2 ist es natürlich denkbar, daß zwei Programme zur selben Zeit Sound spielen wollen. Dies ist leider zur Zeit mit den derzeitigen Sound-Treiber meist noch nicht möglich.

Es gibt auch nur wenige Karten, die die gleichzeitige Wiedergabe von MIDI und digitalem Sound erlauben.

Karten mit dem MWAVE-Chip haben die Möglichkeit, mehr als ein WAV-File gleichzeitig abzuspielen, wobei aber das Abspielen zweier MIDI-Files noch die Rechenleistung der Karte übersteigt.

5.9.7 Interfaces

Viele moderne Soundkarten besitzen eine Schnittstelle für ein oder mehrere CD-ROMs. Zum Betrieb dieses CD-ROM muß natürlich wieder ein Treiber geladen werden. Unter OS/2 benutzt man aber besser ein SCSI- oder ATAPI-CD-ROM. Man kann also vom Neukauf eines CD-Laufwerks, das an die Soundkarte angeschlossen wird, nur abraten.

5.9.8 Interrupts

Soundkarten benutzen zum Betrieb einen Interrupt. Einfache Karten wie z.B. die 8-Bit-Soundblaster erfordern nur einen Interrupt (meist »5«). Die komplexeren Soundkarten benutzen oft zwei oder mehr Interrupts (und meist noch einen, sofern das CD-ROM Interface benutzt wird).

Die miroCONNECT-34-Karte benutzt sogar bis zu vier Interrupts, einen davon für das eingebaute Modem.

Deshalb sollte sichergestellt sein, daß genügend freie Interrupts im System zur Verfügung stehen, wenn eine Wavetable-Karte mit vielen Features benutzt werden soll.

Ebenso benutzen die Soundkarten einen Adreßraum für I/O- und DMA-Kanäle. Dieser muß natürlich ebenfalls angegeben werden und im OS/2-System noch zur Verfügung stehen (manchmal muß das Bios des Rechners entsprechend eingestellt werden).

Viele neue Soundkarten benutzen Software, um die Interrupts und DMA-Adressen einzustellen. Diese werden dann unter OS/2 meist vom Treiber der Soundkarte initialisiert.

Theoretisch sollten PCI-basierende Soundkarten nicht mehr die IRQ-Beschränkungen der ISA-Karten aufweisen. Tatsächlich jedoch gibt es noch kaum Soundkarten, die den PCI-Bus benutzen.

ASUS stellt beispielsweise ein Board her, in das man eine spezielle Vibra-16-Karte stecken kann. Ob es hierfür allerdings schon OS/2-Treiber gibt, ist bislang noch nicht geklärt.

5.9.9 Huckepack-Adapter (Daughterboards)

Einige FM-Synthese-Karten haben ein Steckplatz für ein »Wavetable Daughter Board«, das ein einfaches Aufrüsten auf eine Wavetable-Karte ermöglicht.

Dieser Wavetable-Adapter wird von OS/2 oft nur dann erkannt, wenn ein zusätzlicher Schalter bei der Treiber-Initialisierung angegeben wird (bei der Soundblaster mit Wavetable-Board muß beispielsweise ein »/EXT« angegeben werden).

5.9.10 Treiber-Installation

Die Installation von Soundkarten-Treibern kann manchmal recht tückisch sein, besonders, wenn man eine alte Soundkarte durch eine neue ersetzen will. Das Problem in diesem Fall ist, daß die Installation der neuen Treiber oft nicht die alten Treiber vollständig löscht. Das Ergebnis ist dann oft keine Soundausgabe.

Deshalb gibt es einige einfache Schritte, um dies zu vermeiden:

– Alle Einträge auf das \MMOS2-Verzeichnis löschen. Dies sind alle PATH- und LIBPATH-Einträge in der Config.sys.

– Das System neu booten.

– Das komplette \MMOS2-Verzeichnis löschen. Falls es sich nicht löschen läßt, gibt es irgendwo noch andere Verweise auf dieses Verzeichnis, hier kann man dann von Diskette booten und das Verzeichnis \MMOS2 löschen.

– Dann über das Software-Setup von OS/2 die jeweilige Unterstützung der Soundkarte auswählen. Falls die Treiber der Soundkarte nicht im Basispaket aufgeführt sind, sondern nur als Zusatzdiskette verfügbar sind, ist nur das Basis-Multimedia zu installieren.

– Jetzt rebooten.

– Falls Treiber installiert werden sollen, die nicht im Lieferumfang von OS/2 enthalten sind, kann man dies nun tun.

Falls nun alles richtig ist, sollte auch der Startsound von Warp beim Booten zu hören sein.

Bisher ist leider noch davon abzuraten, das Tool *Selektives Löschen* für die De-Installation des MMPM/2 zu verwenden, es sei denn, man bearbeitet hinterher nochmals seine Config.sys-Datei von Hand nach.

5.10 Soundkarten im Überblick

5.10.1 Preiswerte und einfache Karten

SoundBlaster 8 Bit

Die SoundBlaster 8 Bit (z.B. die 8-Bit-Value-Edition) kann Wav-Files in der niedrigsten Qualität abspielen, hat eine akzeptable FM-Synthese, die für die Systemsounds und manche Spiele genügt. Diese Karte ist also die absolute Minimalanforderung, wenn minimale Soundausgabe genügen soll. Für die Spracherkennung ist diese Karte nicht geeignet.

Auf dem OPTi 928 basierende Karten

Diese Karten können Wav-Files in 16 Bit wiedergeben, und haben eine bessere FM-Synthese als die 8-Bit-Karten. OS/2-Treiber sollte es in guter Qualität für diese Karten geben. Manche dieser Karten besitzen eine Aufrüstoption auf eine Wavetable-Karte, wobei hier nicht bekannt ist, ob diese nachher unter OS/2 gut funktioniert

Auf der Aria-16 basierende Karten

Auch diese Karten können Wav-Files in 16 Bit wiedergeben. OS/2-Treiber sollte es in guter Qualität für diese Karten geben.

5.10.2 Gute 16-Bit-Wav-Wiedergabe und gute FM-Synthese

Creative Labs SoundBlaster-Serie

Hier gibt es verschiedene Karten mit unterschiedlichen Adaptern (für CD-ROMs). Alle besitzen gute OS/2-Treiber, sowie eine gute Unterstützung für DOS-Spiele. Man sollte jedoch keine »Plug-und-Play«-Soundblaster kaufen, wenn man nur OS/2 2.x einsetzt.

Für die Soundblaster Pro gibt es eine Wavetable-Erweiterung als Nachrüstsatz. Auch diese Zusatzoption wird von den OS/2-Treibern unterstützt.

Opti 928

Diese Karten sollten ein wenig günstiger sein als die CL-Karten. Bei diesen Karten ist allerdings auch nicht immer klar, ob alle Modelle einwandfrei unter OS/2 laufen.

5.10.3 Gute OS/2-Wavetable-Unterstützung

CL Soundblaster AWE (32) Serie

Diese Karten sollten im allgemeinen sehr gut mit OS/2 funktionieren. Falls es mit diesen Karten Probleme gibt, ist meist ein Konfigurationsfehler schuld. Insbesondere, wenn man die

neuen AWE-32-Treiber nachinstalliert, kann es sein, daß die Pfadangaben manuell zu korrigieren sind, da sonst einige *.DLL-Dateien, die zum Betrieb nötig sind, eventuell nicht gefunden werden.

Diese Karte sollte problemlos mit allen DOS-Spielen zusammenarbeiten, auch bietet sie unter OS/2 hervorragenden Wavetable-Support und gute MIDI-Wiedergabe.

Mit den neuesten Treibern ist es auch möglich, eigene Wavetable-Sounds in das Sound-RAM zu laden, sowie die Instrumente für MIDI besser auszuwählen.

Ebenso problemlos funktioniert diese Karte mit den Warp V4-Neuerungen wie RTMidi und Dart.

SoundBlaster 16 mit Wavetable-Option und einer separaten Wavetable-Erweiterung

Hier gilt im Prinzip dasselbe wie für die AWE32-Karten, vorausgesetzt, man benutzt ebenfalls die neuesten CL-Treiber für diese Karte. Hier muß allerdings in der Config.sys noch der Schalter »/EXT« angegeben werden, damit diese Karte mit der Wavetable-Erweiterung arbeiten kann.

Prometheus Aria16 oder andere Aria-Karten

Für diese Karten gibt es Treiber, die auch Wavetables unterstützen, allerdings können diese mitunter schwierig zu finden sein.

IBM Mwave-basierte Karten (z.B. Miro)

Diese Karten sollte inzwischen ebenfalls problemlos unter OS/2 funktionieren, mit guter Wavetable- und MIDI-Wiedergabe.

Advanced Gravis UltraSound (GUS)

Mit den neuesten »Manley«-Treibern sollten diese Karten problemlos unter OS/2 funktionieren (inclusive Wavetable- und MIDI-Unterstützung) und auch gute Funktionalität für DOS- und WinOS/2-Programme bieten.

Da die Treiber für diese Karte Shareware sind und von einem einzigen Autor geschrieben werden, muß bisher noch mit Einschränkungen in der Treiberfunktionalität gerechnet werden.

So ist es bisher nicht möglich, »aufzunehmen« (RECORD), auch gibt es bisher noch keine Unterstützung für das Wavetable-Feature der neuesten Plug-und-Play-GUS-Karten. Dies kann sich natürlich in Zukunft ändern.

Logitech SoundMan Wave

Hierfür gibt es Shareware-Treiber, allerdings ist über deren Zustand – und die Funktionsweise der Karten unter OS/2 nicht viel bekannt.

5.10.4 Welche Karten man möglichst vermeiden sollte

Karten ohne OS/2-Treiber

Im allgemeinen natürlich Karten, für die der Hersteller angibt, keine OS/2-Treiber herzustellen. Obwohl manche Karten dennoch mit Treibern anderer Hersteller funktionieren, kann es sein, daß nicht die volle Funktionalität der Karte unterstützt wird.

Generelle »Sound-Blaster-kompatible« Karten

Diese »SB-Clones« finden sich überall (oft in billigen Komplettsystemen großer Computerhandelsketten).

An diesen Karten wird man im allgemeinen sehr wenig Freude haben, da diese mit der nächsten Softwaregeneration meist nicht mehr unterstützt werden (wenn sie denn je mit allen Funktionen richtig liefen).

Ensoniq SoundScape (auch Reveal SC600)

OS/2-Wav-Treiber gibt es zwar, jedoch gibt es einige Probleme mit DOS- und Windows-Programmen unter OS/2.

MediaVision Jazz 16

OS/2 kommt zwar mit Treibern für diese Karten, jedoch gibt es bisher noch zu viele Probleme mit der Soundunterstützung für diese Karte.

MediaVision Pro 3D

Ähnlich wie MediaVision Jazz 16, zusätzlich jedoch funktioniert der Wavetable-Support unter OS/2 nicht.

6 Spracherkennung mit VoiceType

von Herwig Diessner

Die Spracherkennung auf einem Computer ist für viele eines der faszinierendsten Gebiete der heutigen Computer-Technologie. Die Fortschritte, die in den letzten Jahren in diesem Bereich erzielt wurden, sind so gewaltig, daß inzwischen marktfähige Produkte etabliert sind. Fast jeden Monat sind in den Fachzeitschriften der Computerbranche mehr oder weniger ausführliche Artikel über die Art, »mit dem Computer zu sprechen«, zu finden. Die Akzeptanz bei den Zielgruppen variiert jedoch stark zwischen euphorischer Begeisterung und strikter Ablehnung derartiger Systeme. Dies konnte ich vor allem bei meinen Produktpräsentationen in großen Firmen oder mehr noch auf diversen Fachmessen feststellen.

In den letzten Jahren wurden sehr wenige Fachbücher über die Spracherkennung veröffentlicht. Dies liegt wohl vor allem daran, daß Firmen ihre neuen Forschungsergebnisse und Technologien im Bereich der Spracherkennung nicht der Öffentlichkeit und damit eventuellen Mitbewerbern preisgeben möchten.

Dieser Abschnitt des Buches ist in zwei Teile gegliedert.

Im ersten Teil wird zunächst der Begriff der automatischen Spracherkennung erörtert, ebenso werden ihre grundlegenden Probleme aufgezeigt. Im Anschluß an die Entwicklungsgeschichte und die mathematischen Grundlagen für Spracherkennungssysteme wird der technische Erkennungsprozeß speziell für das IBM-VoiceType-Diktiersystem dargestellt. Ebenso wird der Arbeitsablauf bei den Anwendern und die daraus resultierenden Veränderungen durch den Einsatz von Spracherkennungssystemen beleuchtet und der praktische Wert der isolierten Spracherkennung erörtert. Anschließend werden einige Gedanken über die möglichen Zukunftsperspektiven für Spracherkennungssysteme angestellt.

Der zweite Teil des Buchabschnittes beschäftigt sich mit dem praktischen Einsatz von VoiceType unter OS/2 Warp 4. Dabei werden die Bereiche Training, Navigation und Diktieren behandelt, sowie die Erstellung von Makros und Vorlagen erklärt.

Den weitaus größten Teil dieses Buchabschnittes habe ich mit dem VoiceType-Diktiersystem erfaßt und konnte somit weitgehend auf die Tastatur meines PCs verzichten.

6.1 Was ist automatische Spracherkennung?

Die automatische Spracherkennung, kurz ASE genannt, ist ein bedeutsames und interdisziplinäres Arbeitsgebiet. Interdisziplinär deshalb, weil neben der Programmierung und der Computertechnik auch die Phonetik, die Linguistik, die Mustererkennung und die künstliche Intelligenz, um nur einige Beispiele zu nennen, an dem Prozeß der Entwicklung einer Spracherkennungslösung beteiligt sind.

Automatische Spracherkennung ist im weitesten Sinne die Simulation der menschlichen Spracherkennung durch Maschinen. Mit einem automatischen Spracherkenner wird also versucht, das menschliche Sprachverstehen nachzuahmen. Bereits seit vielen Jahren wurde intensive Forschung betrieben, um in der Praxis einsetzbare Spracherkennungssysteme zu entwickeln. Inzwischen sind verschiedene Spracherkennungssysteme auf dem Markt etabliert und erweitern die Möglichkeiten der Mensch-Maschine-Kommunikation. Zweckmäßigerweise wird zur Spracherkennung in der Regel ein Computer, inzwischen genügt schon ein PC, verwendet. Neben Tastatur und Maus dient nun das Mikrophon als dritte direkte Eingabemöglichkeit dazu, Daten in den Computer einzugeben.

6.1.1 Klassifizierung von Spracherkennungssystemen

Bei der ASE handelt es sich nicht um ein einheitliches Spracherkennungssystem, es lassen sich viel eher diverse Spracherkennungssysteme für unterschiedliche Einsatzmöglichkeiten unterscheiden. Abbildung 6.1 gibt einen Überblick, wie sich die automatischeSpracherkennung differenzieren läßt:

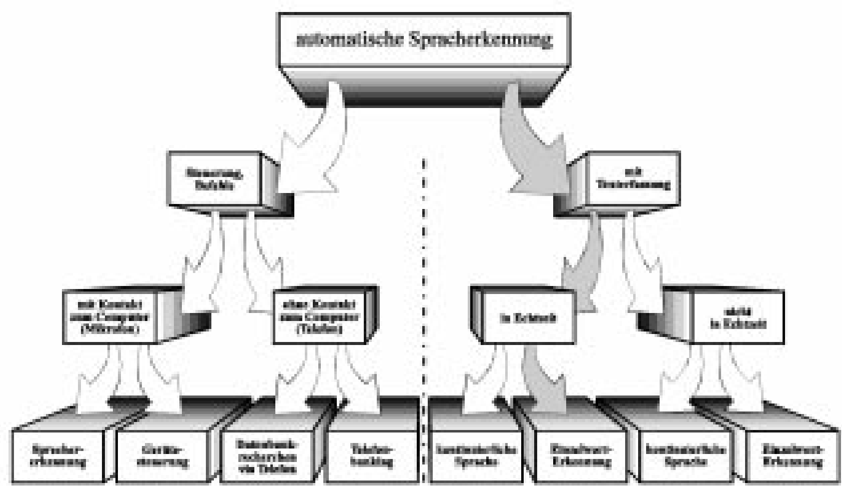


Abb. 6.1: Klassifikation automatischer Spracherkennungssysteme

Wie aus Abbildung 1 ersichtlich wird, unterteilen sich die Spracherkennungssysteme in zwei grundsätzlich verschiedene Gruppen: Die eine Gruppe dient zur Steuerung von Maschinen durch sprachlich gegebene Befehle, die Spracherkennung im engeren Sinne wird zum Erfassen von geschriebenem Text eingesetzt.

ASE zur Steuerung und Befehlseingabe

Zum einen gibt es Spracherkenner, die »nur« zur Steuerung von Maschinen oder zur Befehlseingabe benutzt werden. Das »nur« soll hier nicht auf eine Geringschätzung hindeuten, im Gegenteil: Diese Systeme spielen im Bereich der Serviceleistungen von Firmen und Banken eine immer größer werdende Rolle. Da hier jedoch mehr auf die Systeme mit Texterfassung eingegangen werden soll, erfolgt nur ein kurzer Überblick.

Derartige Steuerungssysteme verfügen in der Regel über einen stark beschränkten Wortschatz, da meist nur wenige Befehle zur Steuerung benötigt werden. Das wichtigste Unterscheidungskriterium bei dieser Art der Spracherkennung ist vor allem die Frage, ob direkter Kontakt zum spracherkennenden Computer bestehen muß, ob man also direkt mit einem Mikrophon die Befehle eingibt, oder ob die Eingabe indirekt, zum Beispiel über ein Telefon, möglich ist. Durch die Beantwortung dieser Frage werden bereits unterschiedliche Anforderungen an das Spracherkennungssystem gestellt: Während die Eingabequalität der menschlichen Stimme, die mit dem Mikrophon in das System eingegeben wird, sehr hoch sein kann (Frequenzbereich von 100 bis 12.000 Hz), leidet die akustische Qualität der gesprochenen Information durch die geringe Bandbreite des Telefonnetzes (300-3400 Hz) in nicht geringem Maße. Ein Erkennungssystem, das also per Telefon bedient wird, muß eine wesentlich höhere Erkennungsleistung bringen, da die zur Verfügung stehenden akustischen Werte wesentlich schlechter sind. Mögliche denkbare Einsatzmöglichkeiten für Sprachsteuerungen mit Kontakt zum Computer sind zum Beispiel Steuerungssysteme, die mittels sprachlich gegebener Befehle Geräte und Maschinen bedienen bzw. steuern. Des weiteren gibt es Systeme, die durch die akustische Analyse der Spracheigenschaften des Sprechers eine Sprechererkennung durchführen und somit zum Beispiel in sicherheitsrelevanten Bereichen eine persönliche Geheimzahl ersetzen können. Solche Zugangskontrollen können in militärischen oder Sicherheitsbereichen von Firmen, wie zum Beispiel in Entwicklungsabteilungen durchgeführt werden. Auf der CeBIT 1995 in Hannover wurde ein System vorgestellt, das z.B. für private Haushalte den Haustürschlüssel überflüssig macht: Jeder, der die Türe öffnen möchte, muß einen bestimmten Satz bzw. ein bestimmtes Wort sprechen. Mit dem analysierten Frequenzspektrum des Sprechers wird der Zugang zum Haus gewährt oder verweigert. Dabei spielt es keine Rolle, ob der Sprecher zum Beispiel heiser oder erkältet ist, das System ist gegenüber derartige Veränderungen in der Stimme unempfindlich.

Spracherkennungssysteme, bei denen kein direkter Kontakt zum Computer bestehen muß, finden vor allem Anwendung im Bankensektor. Dort wird sogenanntes »Telefon-Banking« angeboten, mit dem die Kunden ihre Bankgeschäfte per Telefon erledigen können. Durch sehr wenige Steuerbefehle kann eine sehr gute Erkennung erreicht werden. Meist werden nur die Ziffern von 0 bis 9, »ja« und »nein«, »Hilfe« und »Ende« als Befehle akzeptiert. Auch sind Datenbankrecherchen via Telefon möglich, die bekannteste Anwendung in diesem Bereich ist die telefonische Zugauskunft der Deutschen Bahn AG.

ASE zur Texterfassung

Zum anderen gibt es Systeme, die die gesprochenen Wörter in geschriebenen Text umwandeln können. Der Vorgang, durch den die gesprochenen Wörter in Buchstaben und Symbole umgewandelt werden, wird als »Verschriftung« bezeichnet. Diese Systeme müssen in der Regel über einen sehr großen Wortschatz verfügen, damit zufriedenstellende Erkennungsergebnisse erreicht werden können. Innerhalb dieser Gruppe kann differenziert werden, ob ein System den Text in Echtzeit erfassen kann, oder ob in einem sogenannten »Stapel-Betrieb« (Batch-Job) der Vorgang der Umwandlung von Sprache in Text erfolgen muß. Systeme, die in Echtzeit arbeiten, können den Text unmittelbar nachdem er gesprochen wurde auf dem Bildschirm des Spracherkennungscomputers anzeigen.

Die dritte Ebene der Unterscheidungskriterien befaßt sich mit der Frage, ob eine isolierte Sprechweise, meist jedes Wort für sich gesprochen, zur Spracherkennung erforderlich ist, oder ob der Text kontinuierlich gesprochen werden kann, wie es beim normalen Sprechen üblich ist.

Das in diesem Buch betrachtete System IBM VoiceType arbeitet in Echtzeit und mit isolierter Sprechweise, denn für den täglichen Einsatz am PC werden hauptsächlich Spracherkennungs-Unixsysteme Einsatz finden, die gesprochene Wörter in Schrift umwandeln können.

6.1.2 Systeme für isoliert gesprochene Sprache

Seit einigen Jahren existieren bereits Systeme, die mit dem Prinzip der isolierten oder auch »diskreten« Worterkennung arbeiten. Einen Überblick über den derzeitigen Entwicklungsstand dieser Systeme bietet Abbildung 6.2.

Systeme mit einem begrenzten Wortschatz von ca. 1000 Wörtern werden für den täglichen Einsatz am PC oder am Arbeitsplatz eines Anwalts oder Mediziners kaum geeignet sein. Hier gilt es, ein möglichst umfangreiches und auf den speziellen sprachlichen Kontext abgestimmtes Vokabular zur Verfügung zu stellen. Derartige Systeme sind zur Zeit als sprecherabhängige Spracherkennungslösungen mit einem Wortschatz von bis zu 60.000 Wörtern verfügbar. Ein sprecherunabhängiges Erkennungssystem, das in Echtzeit arbeitet, ist mit dem VoiceType-Diktiersystem unter OS/2 Warp 4 erstmals verfügbar.

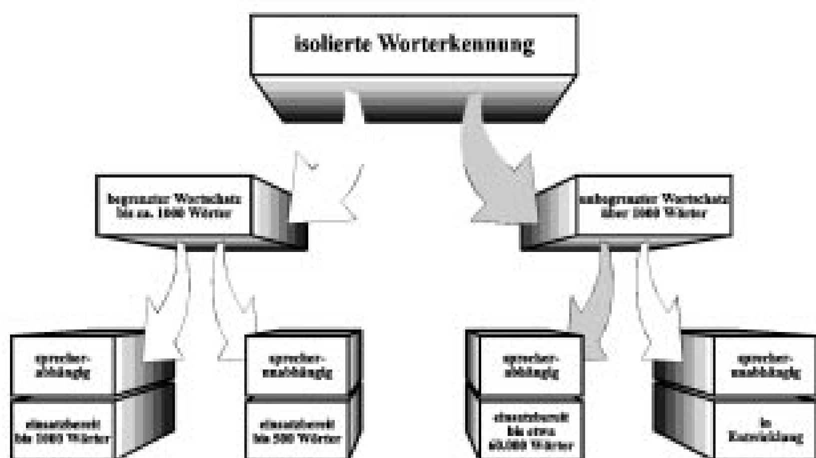


Abb. 6.2: Derzeitiger Entwicklungsstand von Systemen für isoliert gesprochene Sprache

6.1.3 Systeme für kontinuierlich gesprochene Sprache

Wenn man die Tonaufzeichnung eines normal gesprochenen Satzes genauer analysiert, wird man schnell feststellen, daß beim Sprechen fast alle Wörter lückenlos aneinandergereiht wurden. Beim Hören fällt es dem Menschen leicht, die einzelnen Wörter zu unterscheiden. Um wieviel schwieriger muß es für eine Maschine sein, einen solchen zusammenhängenden Redefluß zu strukturieren und in einzelne Wörter zu zerlegen. Man kann sich dies sehr drastisch vor Augen führen, wenn man versucht, einen Text zu lesen, der ohne Leerzeichen und Interpunktion geschrieben wurde.

»Anhanddiesesbeispielskönnensiesichsichervorstellenwieschwierigesfüreinenaseseinmußeinenfließendgesprochenentextzuerkennendennermußbereitswährenddesdiktatsdieeinzelnenwörterrichtigtrennen«

Hieraus ist ersichtlich, selbst Menschen mit jahrelanger Erfahrung im Umgang mit Sprache, haben an solchen Stellen Probleme, den Inhalt des Textes sofort richtig zu erfassen.

Abbildung 6.3 zeigt den Unterschied der Frequenzdiagramme für den nicht kontinuierlich (1) und den kontinuierlich (2) gesprochenen Satz »Die Sonne lacht«.

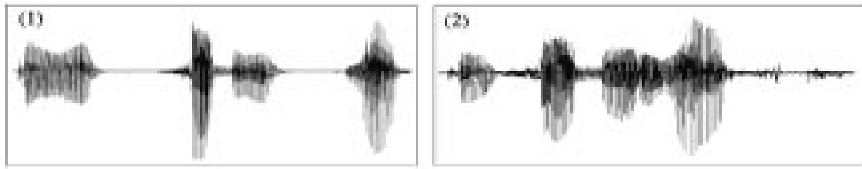


Abb. 6.3: Frequenzdiagramme für diskrete (1) und kontinuierliche (2)

Sprechweise

Das System muß in Echtzeit entscheiden, an welcher Stelle Wörter zu Ende sind. Der ohnehin schon sehr hohe Rechenaufwand bei Systemen für isoliert gesprochene Sprache wird daher um ein Vielfaches übertroffen bei Systemen, die kontinuierlich gesprochene Sprache erkennen sollen. Besonders schwierig wird es, wenn sich im Text zusammengesetzte Wörter befinden. Sollen derartige Komposita richtig erkannt werden, muß das Spracherkennungssystem zusätzlich den Kontext prüfen können, das heißt, die Position des zu erkennenden Wortes im Umfeld der benachbarten Wörter berücksichtigen. Noch besser wäre sogar eine vollständige Grammatikprüfung, die zusätzlich die Funktion des

Wortes im Satz und seine grammatikalische Bedeutung ermitteln kann. Bisher sind jedoch Grammatikprüfungen noch in keinem Produkt realisiert worden.

Aus diesen Gründen existieren bisher nur Prototypen für Spracherkennungssysteme, die kontinuierlich gesprochene Sprache in Echtzeit erkennen können. In der Times vom Freitag, 10.2.1995 wurde über einen Wettbewerb berichtet, an dem verschiedene Universitäten und Firmen teilnahmen, die an kontinuierlich erkennenden Systemen arbeiten. Dort wurden die damaligen Prototypen auf ihre Leistungsfähigkeit hin getestet und verglichen. Auf den ersten Platz gelangte ein an der Cambridge Universität entwickeltes Produkt, das sieben von hundert diktierten Wörtern falsch erkannt hatte. Auf Platz zwei befand sich ein System von IBM mit 8,6 Fehlern, das drittplazierte System produzierte 9,2 Fehler. Alle anderen Teilnehmer wiesen Fehlerraten von über 10% auf. Im Gegensatz dazu erreichen derzeit verfügbare Systeme für die isolierte Sprechweise Erkennungsraten von bis zu über 98%. Man sieht also an diesem Vergleich, daß die Erkennung kontinuierlich gesprochener Sprache bei weitem noch nicht die Erkennungsleistung erreicht, wie die bisher auf dem Markt befindlichen Systeme für die nicht kontinuierliche Sprechweise.

6.1.4 Isolierte Worterkennung und kontinuierliche Spracherkennung

Oft werden falsche Vermutungen angestellt über die Unterschiede der isolierten Worterkennung und der kontinuierlichen Spracherkennung. Deshalb soll hier eine kurze Abgrenzung als Grundlage für das weitere Verständnis durchgeführt werden.

Die akustischen Signale, die durch das Mikrophon aufgenommen werden, müssen durch den Spracherkenner so verarbeitet werden, daß als Ergebnis ein geschriebener Text vorliegt. Die

Verbindung der Akustik mit dem Text wird durch sogenannte Referenzmuster hergestellt. Ein Referenzmuster stellt eine unteilbare Einheit dar, der ein beliebiger Text fest zugeordnet ist. Ein Spracherkennungssystem besitzt einen großen Vorrat an derartigen Referenzmustern, die sozusagen als »Schablone« für die akustisch aufgenommenen Wörter dienen. Eine gesprochene akustische Einheit kann grammatikalisch gesehen ein Wort, eine Phrase oder einen ganzen Satz darstellen. Den Referenzmustern sind dann entsprechend ein Wort, eine Phrase oder ein ganzer Satz zugeordnet. Wird durch den Erkenner jeder akustischen Einheit genau ein Referenzmuster zugeordnet, so spricht man von isolierter Worterkennung. Abbildung 4 zeigt, wie ein isolierter Spracherkenner versucht, dem kontinuierlich gesprochenen Satz »Die Sonne lacht« genau ein Referenzmuster (in diesem Fall für das Wort »die«) zuzuordnen. Da es keine vollständige Kongruenz der Frequenzdiagramme gibt, wird der Satz nicht erkannt.

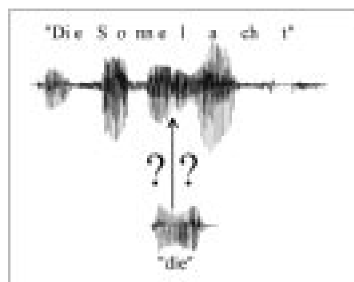


Abb. 6.4: Zuordnung eines Referenzmusters an eine akustische Einheit (isolierte Erkennung)

Beziehen sich ein oder mehrere Referenzmuster auf eine akustische Einheit, so wird dies als kontinuierliche Spracherkennung bezeichnet. Abbildung 6.5 stellt das Prinzip dieser Technik dar: Dem kontinuierlich gesprochenen Satz »Die Sonne lacht« werden drei Referenzmuster zugeordnet. Dadurch wird der Satz erkannt.

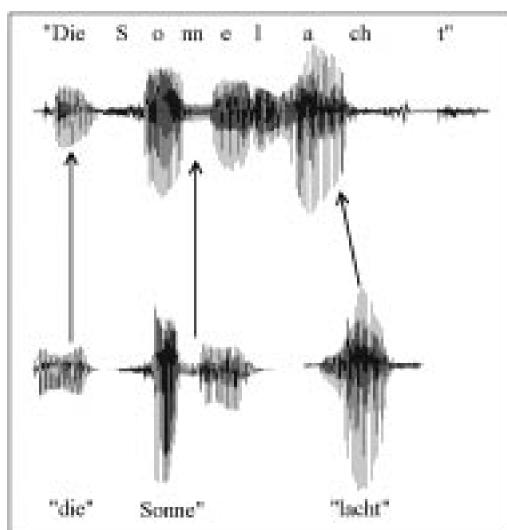


Abb. 6.5: Zuordnung dreier Referenzmuster an eine akustische Einheit (kontinuierliche Erkennung)

Wenn ein Spracherkennungssystem also nach der isolierten Sprechweise arbeitet, so bedeutet dies nicht automatisch, daß man jedes Wort einzeln und für sich sprechen muß. Wenn zum Beispiel ganze Sätze als jeweils ein Referenzmuster hinterlegt sind, so wäre das System in der Lage, diese ganzen Sätze zu erkennen. Folglich müßte zu jedem Satz ein Referenzmuster hinterlegt werden. Erkennt ein solches System den Satz »Die Sonne lacht.«, weil es ein passendes Referenzmuster gefunden hat, so könnte es dennoch den Satz »Die Anne lacht.« nicht erkennen, obwohl nur eine geringe lautliche Abweichung besteht.

Bei der Unmenge von verschiedenen Sätzen ist die technische Realisierung eines solchen Systems mit den heutigen technischen Gegebenheiten unmöglich. Deshalb arbeiten in der Praxis die meisten Systeme, die das Prinzip der isolierten Worterkennung verwenden, so, daß jedes Wort getrennt gesprochen wird. Dadurch wird die Anzahl der zu hinterlegenden Referenzmuster in Grenzen gehalten.

OS/2 Warp 4 kann mit kontinuierlich gesprochenen Befehlen gesteuert werden (Navigationsfunktion). Dies ist möglich, weil der Wortschatz für die Steuerung relativ klein ist und so zum ersten Mal in einem breit verfügbaren Produkt eine kontinuierliche Spracherkennung realisiert werden konnte. Hier werden also einem Satz mehrere Referenzmuster zugeordnet.

6.1.5 Sprecherunabhängigkeit

Ein sprecherunabhängiges System kann von jeder beliebigen Person ohne vorheriges Training benutzt werden. Vor allem bei Sprachsteuersystemen, die per Telefon bedient werden, ist die Sprecherunabhängigkeit zwingende Voraussetzung für die Praxistauglichkeit. Bei Systemen, die zur Steuerung oder Befehlseingabe mit nur sehr wenigen Wörtern im aktiven Wortschatz auskommen, ist eine Sprecherunabhängigkeit leichter zu realisieren als bei Spracherkennern, die zur Mengentexterfassung dienen sollen.

Bei Texterfassungssystemen muß differenziert werden, ob sich die Sprecherunabhängigkeit auf den Wortschatz oder auf das gesamte Spracherkennungssystem bezieht. Bei einem Spracherkennungssystem ohne mitgeliefertem sprecherunabhängigen Wortschatz muß jeder Benutzer, der mit dem System arbeiten möchte, seinen eigenen, persönlichen Wortschatz erst aufbauen. Aus jedem Wort, das der Benutzer diktiert, muß der Rechner ein Referenzmuster erzeugen. Diese Arbeit muß jeder Benutzer auf sich nehmen, da er ja auf keinen vorhandenen Wortschatz zurückgreifen kann. Erst wenn der selbst aufgebaute Wortschatz repräsentativ genug ist, können ausreichend gute Erkennungsergebnisse erreicht werden. Dadurch wird für jeden Benutzer die Einarbeitungszeit sehr hoch werden und die Produktivität wird für lange Zeit stark absinken. Ein mitgelieferter sprecherunabhängiger Wortschatz ist somit sehr sinnvoll, da dieser nur noch auf die Sprechweise des Benutzers angepaßt und nicht erst eingegeben werden muß. Die am häufigsten verwendeten Wörter sind in solch einem Wortschatz bereits enthalten und mit Referenzmustern hinterlegt. Diese werden nur noch an den Sprecher adaptiert, zum Beispiel lernt das System, ob der Sprecher »wichtig« oder »wichtich« sagt.

Man stelle sich einmal vor, ein Bayer, ein Sachse und ein Schweizer sollten vom System sofort und ohne Initialtraining erkannt werden. Dies ist selbst für Menschen schwierig, da die regionalen sprachlichen Eigenheiten der Sprecher sehr verschieden sind. Ein System, das nicht

auf die zusätzlichen Kanäle, die dem Menschen zur Verfügung stehen (z.B. Gestik, Mimik), zurückgreifen kann, wäre hier hoffnungslos verloren. Ein solches Anpassungstraining zwischen Mensch und Maschine war bis vor kurzem noch unabdingbar. Diese Trainingsphase für ein Spracherkennungssystem kann, muß aber nicht unbedingt so vonstatten gehen, daß man vor Beginn des Systemeinsatzes vorgegebene Sätze nachsprechen muß. Das Training kann auch sukzessive während der täglichen Arbeit mit dem System erfolgen. Dadurch wird erreicht, daß für den Benutzer die Einarbeitungszeit entfällt, kann aber zur Folge haben, daß die Fehlerrate zu Beginn der Arbeit mit dem Spracherkennungssystem höher liegt als mit einem Initialtraining. Es ist also nicht von maßgeblicher Bedeutung, ob ein Benutzer das System vor Beginn der Arbeit auf seine persönliche Sprechweise trainieren muß oder nicht. Eine vollkommene Sprecherunabhängigkeit wird wohl für längere Zeit nicht erreichbar sein.

Auch die neue Version des VoiceType Diktiersystems, das in OS/2 Warp 4 integriert ist, erreicht nicht die vollkommene Sprecherunabhängigkeit. Der sprecherunabhängige Standardwortschatz wurde jedoch von einer wesentlich größeren Gruppe von Personen als Referenz aufgenommen: Etwa 290 Sprecher, deren Muttersprache durchweg Deutsch (allerdings in unterschiedlichsten Dialekten) war, haben alle Wörter des gesamten Wortschatzes gesprochen. Dadurch wurde eine größere Sprecherunabhängikeit erreicht und die Notwendigkeit eines Initialtrainings vor Verwendung von VoiceType enorm reduziert.

Herbert Bierfert beschreibt die Unmöglichkeit der absoluten Sprecherunabhängigkeit in seiner Arbeit »Automatische Spracherkennung – von der Theorie zur Praxis« folgendermaßen: »Da jeder Sprecher Eigentümlichkeiten in Stimme und Aussprache aufweist, die sich auf den mechanischen Erkennungsvorgang negativ auswirken, muß hier das Gerät sich dem Sprecher anpassen, was nicht für eine unbegrenzte Sprecherpopulation geleistet werden kann. Deshalb ist die Beschränkung auf eine begrenzte Sprecherpopulation unumgänglich.«

Die von Bierfert angesprochenen Eigentümlichkeiten in Stimme und Aussprache sollen im folgenden Kapitel genauer beleuchtet werden.

6.2 Grundlegende Probleme bei der Spracherkennung

6.2.1 Akustische Variationen

Die größte Schwierigkeit bei der automatischen Spracherkennung besteht darin, daß ein und dasselbe Wort nie ein zweites Mal absolut identisch ausgesprochen werden kann, selbst wenn es der Sprecher versucht. Weitere Klangvariationen werden hervorgerufen durch die physischen und psychischen Zustände des Sprechers, durch den sprachlichen Kontext, durch die Eigenschaften des Mikrophons oder Umgebungseinflüsse. Durch Hintergrundgeräusche kann der Frequenzverlauf eines gesprochenen Wortes derart stark verändert werden, daß die Erkennung durch den Computer eingeschränkt oder sogar unmöglich wird. Zur Zeit werden Syste-

me entwickelt, die mit Unterstützung durch eine Videokamera, die auf den Mund des Sprechers gerichtet ist, die Erkennung auch in lauten Umgebungen optimieren sollen.

Zwischen verschiedenen Sprechern zeigen sich natürlich noch gravierendere Unterschiede, abhängig von deren Alter und Geschlecht. Zusätzlich können artikulatorische Eigenheiten bis hin zu Sprachfehlern den Frequenzverlauf der Stimme beeinflussen. Abbildung 6.6 zeigt Frequenzdiagramme des Satzes »Die Sonne lacht«, jeweils von einer anderen Person gesprochen, beide männlichen Geschlechts, gleichen Alters und ohne Sprachfehler. Die Aufnahme fand unmittelbar nacheinander im gleichen Raum, mit dem gleichen Mikrophon statt.

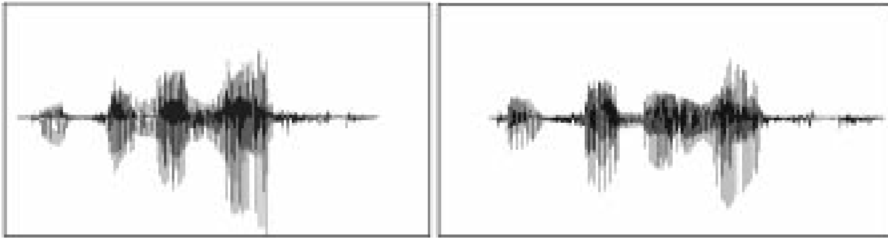


Abb. 6.6: Frequenzdiagramme unterschiedlicher Sprecher

Man kann daraus erkennen, daß selbst bei annähernd gleichen Umgebungsbedingungen erhebliche Unterschiede im Frequenzdiagramm auftreten. Neben der Lautstärke, dem Rhythmus und der Frequenz hat auch die Sprechgeschwindigkeit eine nicht unerhebliche Bedeutung.

6.2.2 Inhaltliches Verstehen und Zusatzwissen

Neben der reinen Akustik, wie die Betonung und die Stimmlage des Sprechers, kann ein Mensch auch aus der Gestik und Mimik seines Gesprächspartners zusätzliche Informationen erhalten, zum Beispiel über seine Gemütslage. Diese Kanäle bleiben einer Maschine natürlich unzugänglich und dies bedeutet ein weiteres Problem für die automatischen Spracherkennungssysteme. Ein weiterer Vorteil des menschlichen Spracherkennens ist, daß ein Mensch neben dem wörtlichen Verstehen auch inhaltlich versteht, um was es sich in dem Gespräch handelt. Psychologen nennen dieses Phänomen den »Cocktail-Party-Effekt«. Bei einer Party leidet die Qualität der Sprache durch Musik und andere Gespräche im Hintergrund. Dennoch können selbst undeutlich gesprochene Wörter vom Menschen richtig erkannt werden, weil er den Inhalt versteht. Man stelle sich zum Beispiel vor, in einem Bus zu sitzen. Selbst wenn der Fahrer sehr undeutlich spricht, ist zu erwarten, daß er den Namen einer Straße oder eines Platzes ansagen wird. Solches Zusatzwissen, das sogenannte Weltwissen, bleibt dem Computer verborgen.

Probleme im deutschen Sprachraum

Speziell im deutschen Sprachraum gibt es noch weitere bedeutsame Probleme, die eine Spracherkennung erschweren.

Aktiver Wortschatz

Während eine englisch sprechende Person einen aktiven Wortschatz von ca. 800 Wörtern besitzt, umfaßt der Wortschatz einer deutsch sprechenden Person etwa 4000 Wörter. Ausnahmen nach oben oder unten gibt es natürlich, so hatte z.B. Goethe einen aktiven Wortschatz von etwa 24.000 Wörtern zur Verfügung, um seine Gedanken auszudrücken. Typisch für die Spracherkennung ist, daß jede Wortform wie ein eigenes Wort gezählt wird. Da ein solches System nur auf lautliche Eigenschaften hin untersucht, gibt jede Flexion eines Wortes einen neuen Eintrag im Wörterbuch der Spracherkennung. Das Wort Baum hat z.B. weitere Formen: Baumes, Bäume und Bäumen. Jede Flexion dieses Wortes, also jede Wortform, benötigt im Vokabular des Spracherkenners einen eigenen Eintrag. Während im Englischen in der Regel jedes Verb nur etwa vier Flexionsformen besitzt, sind es im Deutschen weit über zehn verschiedene Schreibweisen für ein Verb. Allgemein gesehen gibt es für alle Wörter im Englischen 2,2 Flexionen je Grundform, im Deutschen 5 und im Französischen sogar 7. Im deutschsprachigen Raum gibt es bis zu einer Million Wortformen. Der Einfachheit halber wird jedoch bei Spracherkennungssystemen nicht von Wortformen, sondern von Wörtern gesprochen.

Homophone

Ein weiteres Problem für eine Spracherkennung ist das Auftreten von Homophonen, also Wörtern, die gleich gesprochen werden aber unterschiedlich geschrieben werden müssen. So klingen die Wörter »Meer« und »mehr«, »Lärche« und »Lerche«, »viel« und »fiel«, »wieder« und »wider« absolut identisch, ihre Bedeutung ist jedoch unterschiedlich und selbst dem Menschen bereiten diese Wörter immer wieder Probleme bei ihrer Differenzierung. Sätze wie der folgende bringen auch heute noch die meisten Spracherkennungssysteme an die Grenze ihrer Leistungsfähigkeit:

»Der junge Junge fiel viel und fällt noch immer viel auf dem Feld.«

Noch schlimmer wird es bei dem folgenden Satz: »Wenn hinter Fliegen Fliegen fliegen, fliegen Fliegen Fliegen hinterher.«

Derzeit können nur die leistungsfähigsten Spracherkennungssysteme derartige Homophone mit ausreichender Zuverlässigkeit unterscheiden. Wird ein solcher Satz umgestellt (z. B. »Fliegen fliegen hinter Fliegen.«), so kann es sein, daß erneut Erkennungsschwierigkeiten auftreten.

Groß- und Kleinschreibung

Wie der Satz mit den Fliegen unter Punkt 2.3.2 bereits zeigte, können in der deutschen Sprache Wörter völlig unterschiedlichen Sinn erhalten, wenn sie groß oder klein geschrieben werden. Auch dies ist ein Problem für die automatische Spracherkennung, da nur durch Kontextprüfung ein Unterschied zwischen diesen Wörtern festgestellt werden kann. Es handelt sich hierbei also um Sonderformen von Homophonen. Durch die Veränderung der Groß- und Kleinschreibung können sich völlig unterschiedliche Sinnzusammenhänge ergeben: »Der gefangene Floh.« oder »Der Gefangene floh.«. Bei einem solchen Fall kann die Spracherkennung nur dann richtig erkennen, wenn sie sich ständig an den Sprecher und seine Wortwahl anpaßt. Hätte zum Beispiel ein Journalist, der regelmäßig für die Tageszeitung den Polizeibericht erfaßt, diesen Satz gesprochen, so könnte das System aufgrund von Kontextstatistiken

entscheiden, daß es hier »Der Gefangene floh.« ausgeben muß. Bei einem Autor für Kindermärchen wäre die Schreibweise »Der gefangene Floh.« wohl wahrscheinlicher.

Komposita

Zusammengesetzte Wörter, also Komposita, sind ebenfalls typisch für die deutsche Sprache. Man kann fast jedes Wort mit einem anderen kombinieren und der Kreativität der Sprecher oder Schreiber sind fast keine Grenzen gesetzt. Selbst wenn ein Spracherkenner die Wörter Steuer und Lüge kennt, so würde er das Kompositum Steuerlüge beim ersten Diktieren nicht erkennen.

Derivationen

Auch gibt es von verschiedenen Verben fast beliebig viele Derivationen, das heißt, ein Stammwort wird durch Anhängen oder Voranstellen von anderen Wörtern oder Silben in der Sprechweise und im Sinn verändert. Ein Beispiel wäre das Verb »gehen«: »weggehen«, »hingehen«, »umgehen« sind Derivationen, die ein Spracherkenner nicht korrekt erkennen kann, auch wenn sich die Grundform »gehen« in seinem Vokabular befindet.

Steigende Informationsflut

Ein Problem unserer heutigen Zeit ist auch das Anwachsen der Informationsflut. Während die »Zehn Gebote« nur 279 Wörter umfaßten, beinhaltete die Amerikanische Unabhängigkeitserklärung bereits 3000 Wörter. In heutigen Zeiten ist die Informationsflut noch weiter angewachsen: Die EG-Verordnung über den Import von Karamelbonbons zum Beispiel umfaßt stolze 25.911 Wörter. Aus diesem Grunde muß der Wortschatz für ein Spracherkennungssystem ständig erweitert werden. Das System muß immer mehr Wörter verwalten und sich dynamisch anpassen können, um dieser Entwicklung gerecht zu werden.

6.3 Entwicklungsgeschichte der Spracherkennung

Bereits seit über zwanzig Jahren wird in den Labors unterschiedlichster Firmen intensiv an der automatischen Spracherkennung geforscht und entwickelt. Anfängliche Erfolgsmeldungen mußten zum Teil bald wieder eingeschränkt und revidiert werden. Auch die Firma IBM forscht und entwickelt seit langem an Spracherkennungssystemen. Bereits das erste kommerziell genutzte Sprachausgabegerät (noch keine Spracherkennung) war eine Entwicklung von IBM. Dieses Modell 7772 wurde im Jahre 1962 auf den Markt gebracht. 1984 wurde ein Spracherkennungssystem vorgestellt, das mit Hilfe eines Großrechners in einem mehrere Minuten dauernden Rechenvorgang etwa 5000 englische Einzelwörter erkennen konnte.

TANGORA

Wissenschaftler des Forschungslabors der IBM in Yorktown Heights, USA, entwickelten im Jahr 1986 den ersten Prototyp des Spracherkennungssystems TANGORA für Englisch. Der

Name wurde als Hommage an den Weltrekordhalter im Schreibmaschinenschreiben, Alberto Tangora, gewählt. Bei diesem System war es durch spezielle Mikroprozessoren möglich, die komplizierten Verarbeitungsschritte der gesprochenen Sprache auf einem Arbeitsplatzrechner in Echtzeit durchzuführen. Das bemerkenswerte an diesem System war, daß es bereits eine Kontextprüfung beinhaltete. Durch sogenannte Trigrammstatistiken konnte das System Homophone sicher unterscheiden.

Ein Trigramm ist eine Verbindung von jeweils drei Wörtern innerhalb des Textes. Das System führt darüber Statistik, welche Wörter mit welchen anderen in welcher Schreibweise auftreten. Diese Statistiken werden beim Erkennungsprozeß ausgewertet, um die richtigen Wörter in der korrekten Schreibweise zu finden.

Seit 1988 wurde im Wissenschaftlichen Zentrum Heidelberg der Prototyp für die deutsche Version von TANGORA entwickelt. Einsatz fand dieses System in der Medizin zum Diktieren von radiologischen Befunden, im Gerichtswesen für juristische Gutachten und im Büro für die allgemeine Geschäftskorrespondenz. Die einzelnen Vokabulare umfaßten zwischen 20.000 und 30.000 Wörter, wobei, wie bereits erwähnt, jede Wortform als eigenes Wort zählt. Auf der CeBIT 1991 wurde das System erstmals einer breiteren Öffentlichkeit vorgeführt. Damals waren noch spezielle akustische Vorbereitungen notwendig. Das System mußte in einem vom Lärm der Messe abgeschirmten Raum demonstriert werden.

IBM Speech Server Series (ISSS)

Seit Dezember 1992 ist die TANGORA-Technik in einem Produkt unter dem Namen ISSS (IBM Speech Server Series) verfügbar. Zur Spracherkennung ist eine IBM RS/6000, also ein auf RISC-Prozessoren basierendes Computersystem mit dem Betriebssystem AIX, notwendig. Die Implementierung erfolgt als eine Client/Server-Lösung. Die akustische Eingabe des Diktates wird auf einem PC mit OS/2 oder auf einem AIX-Rechner durchgeführt, der Server übernimmt die Umwandlung der Akustik in Text. Besonders in Krankenhäusern und großen Kliniken kommt bisher dieses System zum Einsatz.

IBM Personal Dictation System / IBM VoiceType Diktiersystem

Ende 1993 wurde ein neues Produkt, das auf der TANGORA-Technologie basiert, angekündigt. Der Preis war geradezu eine Sensation. Während alle bisherigen Lösungen über 20.000 DM gekostet hatten, wurde das IBM Personal Dictation System, das kurze Zeit später in IBM VoiceType Diktiersystem umbenannt wurde, für unter 1000 $ auf den Markt gebracht. Dieses System war eine reine PC-Lösung und damit auch für den breiten Massenmarkt bestimmt. Bei der Vorstellung des Systems auf der CeBIT 1994, damals noch als englische Version der Spracherkennung, stieß das System sowohl bei den Messebesuchern als auch von Seiten der Presse auf unglaubliche Resonanz. Wenige Wochen später wurde das Diktiersystem oder auch die »elektronische Sekretärin«, wie die Spracherkennung in der Presse genannt wurde, in der deutschen Version auf den Markt gebracht. Der Trend zur Begeisterung für dieses Produkt wurde auf der CeBIT 1995 fortgesetzt, denn es wurde zum ersten Mal die deutsche Version mit speziellen Fachvokabularen für Mediziner und Anwälte vorgeführt, wobei die Vorführungen selbst in den lautesten Hallen der Messe durchgeführt werden konnten, ohne daß die Erkennungsleistung besonders stark beeinträchtigt wurde.

6.4 VoiceType unter OS/2 Warp 4

Wiederum ein Jahr später wurde auf der CeBIT 1996 VoiceType 3.0 für Windows 95 vorgestellt. Dieselbe Technik ist in OS/2 Warp 4 integriert. Highlights dieser Version sind die kontinuierliche Sprechweise bei der Navigation, der Verzicht auf ein anfängliches Training (»Enrollment«) und der Vorteil, daß keine Spezielle DSP Karte mehr notwendig ist (DSP: digitaler Signal-Prozessor). Eine 16-Bit-Soundkarte genügt für die Spracherkennung. Dafür muß der PC jedoch mindestens mit einem Pentium mit 90 Mhz ausgestattet sein.

6.5 Algorithmen zur Spracherkennung

Die kleinste Einheit bei der Spracherkennung ist das Phonem. »Dies ist die kleinste, bedeutungsunterscheidende, aber doch nicht selbst bedeutungstragende, lautsprachliche Einheit«. Bei normaler Sprechgeschwindigkeit hat ein Phonem die Dauer von etwa 10 bis 40 Millisekunden. Wenn zwei Laute in derselben lautlichen Umgebung vorkommen können, jedoch zu verschiedenen Wörtern gehören, so sind sie zwei verschiedene Phoneme. Zwei Beispiele für verschiedene Phoneme sind das »m« in »mein« und das »d« in »dein«. Abbildung 6.7 zeigt die Frequenzdiagramme der beiden Wörter.

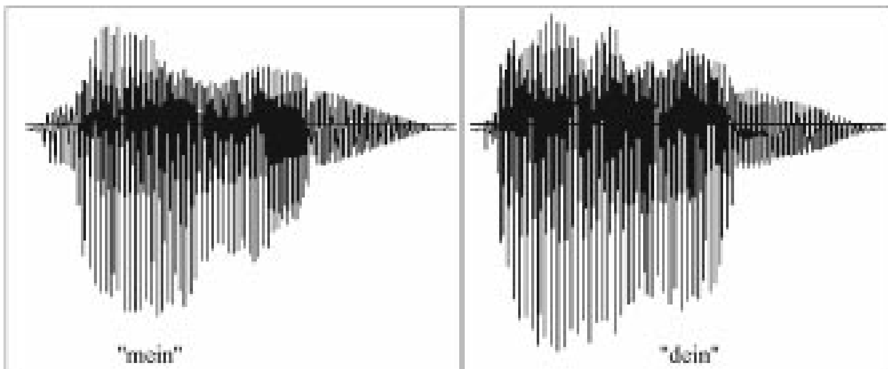


Abb. 6.7: Phonemunterscheidung anhand zweier Frequenzdiagramme

Ein ganzes Wort besteht aus mehreren Phonemen. Bei der Spracherkennung werden in kurzen Abständen, etwa im Abstand von 10 msec, Kurzzeitspektren der Akustik erstellt. Wenn ganze Wörter erkannt werden sollen, wird die Folge dieser Kurzzeitspektren mit Mustervektoren verglichen. Um diese Vergleiche möglichst schnell und optimal für die Erkennung durchzuführen, stehen drei wichtige Verfahren zur Verfügung:

– die dynamische Programmierung,

– die Darstellung in Form von »Hidden-Markov«-Modellen und

– die künstliche Intelligenz

6.5.1 Mustervergleich durch dynamische Programmierung

Würden alle Kurzzeitspektren mit allen Mustervektoren verglichen, wäre dies ein immenser Rechenaufwand. Um die Rechenzeit zu verkürzen und damit die Erkennungsgeschwindigkeit zu steigern, wird das Verfahren der »Dynamischen Programmierung« eingesetzt. Hierbei wird das untersuchte Wort mit Referenzwörtern, die im Wortschatz gespeichert sind, verglichen. Da jedes Wort in unterschiedlicher Geschwindigkeit gesprochen werden kann, muß eine nichtlineare Zuordnung der Kurzzeitspektren zu den einzelnen Mustervektoren erfolgen, das heißt, die Zeitachse wird verzerrt. Über einen Algorithmus wird nun versucht, die Zuordnungen der Mustervektoren zu den Kurzzeitspektren rekursiv zu berechnen. Eine rekursive Problemlösung bedeutet, daß sich zum Beispiel eine Prozedur im Programm selbst erneut aufruft, jedoch mit anderen Parametern. Gleichzeitig wird durch den Algorithmus darauf geachtet, daß die Kurzzeitspektren nur in der richtigen Reihenfolge, also zum Beispiel nicht zeitlich gesehen rückwärts, den Mustervektoren zugeordnet werden können. Dasjenige Referenzmuster, das den kleinsten Abstand zu dem zu untersuchenden Wort aufweist, ist das Erkennungsergebnis.

6.5.2 Erkennung mit »Hidden-Markov«-Modellen

Ein weitaus besseres, aber auch rechenintensiveres Verfahren zur Erkennung von Wörtern basiert auf Markov-Ketten. Das sind Ketten von Übergangswahrscheinlichkeiten von einem Phonem zum nächsten. Hidden-Markov-Modelle benötigen vor dem Einsatz eine gewisse Trainingsphase, in der verschiedene Klassen von Erzeugungsmodellen aufgebaut werden. Nach dem Training wird bei der Erkennung für einen vorliegenden unbekannten Musterverlauf (Wortmuster X mit der Länge T; Darstellung: $\underline{V}X=\{x_t1,x_t2,...,x_tT\}$) die Wahrscheinlichkeit dafür berechnet, daß das Modell diesen Verlauf erzeugen kann. Diese Berechnung wird für jedes Modell aus jeder Klasse durchgeführt. Daraus resultiert der hohe Rechenaufwand, den dieses Verfahren mit sich bringt. Um ein Hidden-Markov-Modell für ein Wort mit n Phonemen und der Länge T vollständig zu berechnen, müssen $2Tn^T$ Berechnungen durchgeführt werden. Da selbst bei relativ kleinen Werten für die beiden Variablen riesige Werte für die Anzahl der Rechenoperationen erreicht werden, kann in der Praxis kein Prozessor ein Hidden-Markov-Modell vollständig berechnen. In der Praxis muß man hier einen Kompromiß schließen. Deshalb werden die Hidden-Markov-Modelle durch spezielle Rechenverfahren abgekürzt. Dadurch wird zwar die Berechnung ungenauer, dafür aber wesentlich schneller, so daß dieses Verfahren auch für Systeme, die in Echtzeit Sprache erkennen, eingesetzt werden kann. Abbildung 6.8 skizziert das Prinzip der Erzeugungsmodelle für die Trainings- und Erkennungsphase.

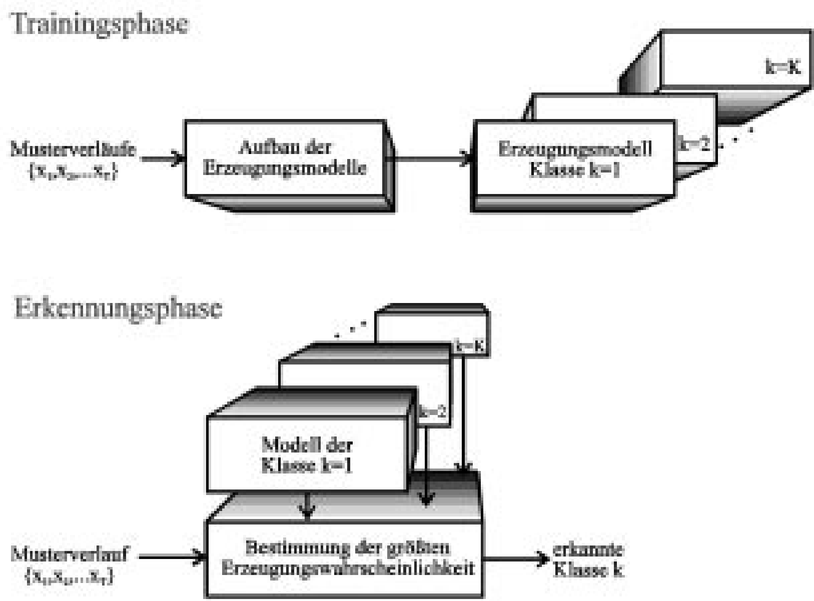


Abb. 6.8: Erzeugungsmodelle für die Trainings- und Erkennungsphase bei HMMs

Die Phoneme eines Wortes werden durch Kurzzeitspektren dargestellt. Jedes Kurzzeitspektrum ist einem Zustand einer Markov-Kette zugeordnet, wobei jeder Zustand i eine Wahrscheinlichkeit $p(\underline{V}x|s=i)$ für das Erzeugen der spektralen Vektoren $\underline{V}x$ enthält. Diese Zustände werden von links nach rechts durchlaufen, wobei die Übergangswahrscheinlichkeiten von einem Zustand zum nächsten berücksichtigt werden.

Ein Modell mit r = 4 Zuständen, denkbar wäre zum Beispiel das Wort »Ofen«, wird in Abbildung 6.9 gezeigt. Die Wahrscheinlichkeit für die Selbstübergänge $a_{t}i_{t}i$ gibt an, daß mehrere Kurzzeitspektren von ein und demselben Phonem (und damit dem Zustand i) erzeugt wurden. Ebenfalls ist es möglich, daß einzelne Zustände übersprungen werden können. Von außen ist es nicht ersichtlich, welcher Zustand aktuell durchlaufen wird. Deshalb heißt dieses Modell auch Hidden-Markov-Modell (hidden: engl. verborgen).

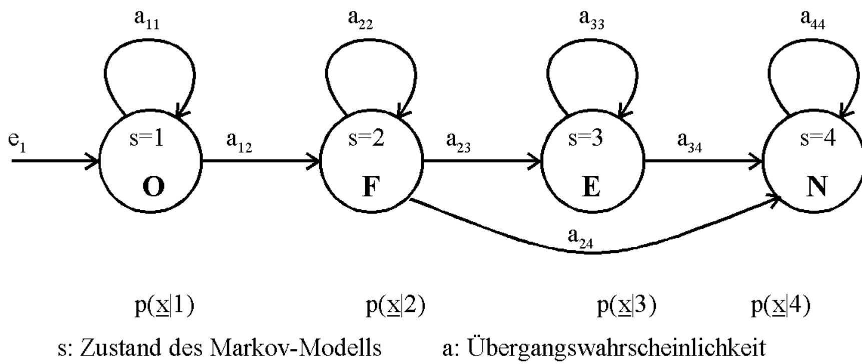


Abb. 6.9: Hidden-Markov-Modell für ein gesprochenes Wort mit vier Phonemen (»Ofen«)

Der Übergang a_t2_t4 von Zustand s=2 auf Zustand s=4 ist dann denkbar, wenn der Sprecher das »e« im Wort »Ofen« nicht ausspricht. Derzeit findet das Prinzip der Hidden-Markov-Modelle in den meisten verbreiteten Spracherkennungssystemen Anwendung.

6.5.3 Methoden der künstlichen Intelligenz

Bei diesem Verfahren werden hauptsächlich neuronale Netze zur Spracherkennung eingesetzt. Diese neuronalen Netze lehnen sich an die Funktion des menschlichen Gehirns an. Eine menschliche Nervenzelle, Neuron genannt, lernt in der Form, daß es über »Brücken«, sogenannte Synapsen, mit anderen Nervenzellen verbunden wird. Diesen Lernvorgang versucht man nun durch entsprechende Programmierung auf den Computer zu übertragen. Dabei werden mehrere Schichten verwendet, um die Erkennungsleistung zu realisieren (Abbildung 10). Die gesprochenen Signale liegen an der Eingangs-Schicht an (hier ein Wortmuster X mit der Länge T = 2; $\underline{V}X = \{x_t1, x_t2\}$). In der verborgenen Schicht werden sie verarbeitet, bis das erkannte Wort an der Ausgangs-Schicht vorliegt (hier zum Beispiel a_t1 bis a_t3). Die Anzahl der verborgenen Schichten kann beliebig variieren. Dabei ist es nicht ersichtlich, welche Vorgänge im Inneren des Systems zur Erkennung ablaufen. Nach dieser Grundstruktur des Multi-Layer-Perceptron (MLP) arbeiten die meisten neuronalen Netze, wie sie auch zum Beispiel für die optische Buchstabenerkennung (OCR) Anwendung finden.

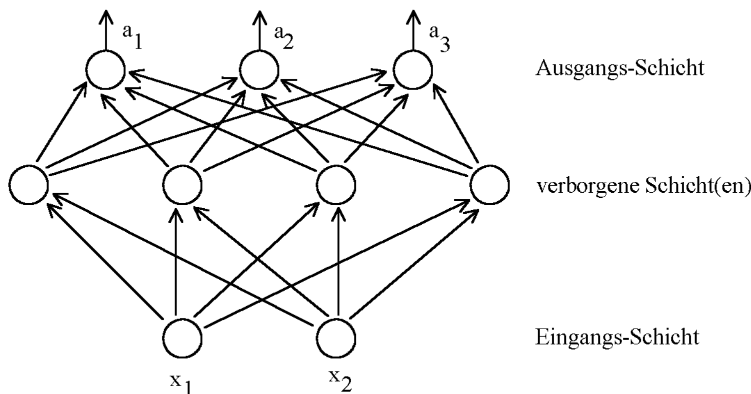


Abb. 6.10: Einfache Struktur für ein Multi-Layer-Perceptron (MLP)

Erkennungsprozeß des IBM VoiceType

Der technische Prozeß der Spracherkennung speziell beim IBM VoiceType-Diktiersystem soll im folgenden anhand des Satzes »Die Sonne lacht.« erläutert werden. Abbildung 6.11 zeigt die Anordnung der einzelnen akustischen Analysen im Dekodierer und den jeweils notwendigen Input.

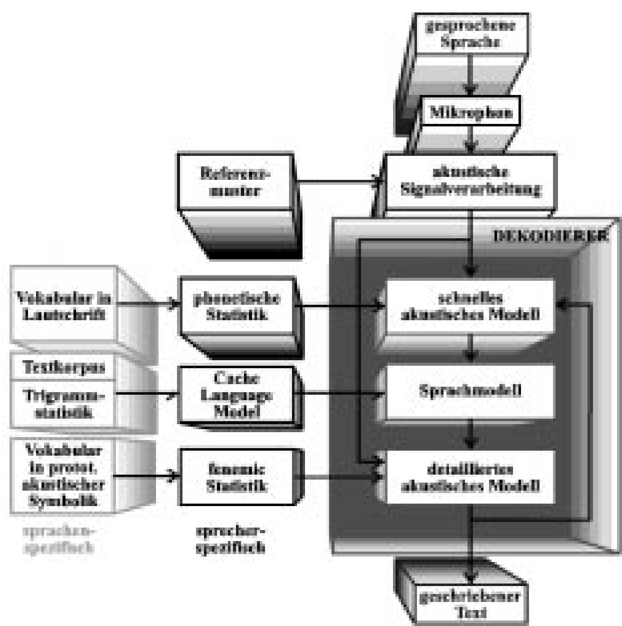


Abb. 6.11: Dekodierprozeß des IBM VoiceType Diktiersystems

6.5.4 Zerlegung der Akustik

Zunächst wird die mit dem Mikrophon erfaßte gesprochene Sprache durch die akustische Signalverarbeitung mittels Fourier-Transformationen in Kurzzeitspektren zerlegt. Bei einer solchen Transformation werden diejenigen Einzelfrequenzen berechnet, aus denen sich ein Kurzzeitspektrum zusammensetzt. Diese Kurzzeitspektren werden zu einem Vektor zusammengefaßt, der mit den Referenzvektoren des Systems verglichen wird. Aus der Folge dieser akustischen Symbole wird die Worterkennung (Dekodierung) durchgeführt.

Dekodierung

Die Dekodierung (vgl. Abbildung 6.11) erfolgt in 3 Stufen. Zunächst wird ein schnelles akustisches Modell verwendet, gefolgt von einer Selektion mit Hilfe des Sprachmodells. Zum Schluß wird durch das detaillierte akustische Modell das endgültige Wort ausgewählt. Abbildung 6.12 zeigt die Wortkandidaten nach den jeweiligen Verarbeitungsstufen.

Erkennungsprozeß		
schnelles akustisches Modell (Wortkandidaten nach Wahrscheinlichkeit)	Nacht lacht macht ... kocht ... freie	1 2 3 ... 43 ... 150
Sprachmodell (Wortkandidaten nach Trigrammen)	Nacht lacht macht ... kocht	1 2 3 ... 14
detailliertes akustisches Modell (Wortkandidaten nach Wahrscheinlichkeit)	lacht Nacht macht	1 2 3

Abb. 6.12:
Wortkandidaten während des Erkennungsprozesses

Schnelles akustisches Modell

In diesem ersten Schritt werden die Wörter selektiert, die die größte Wahrscheinlichkeit für das richtige Wort besitzen. Die Anzahl dieser Wortkandidaten beträgt ca. 150. Dabei wird für jedes Wort die Wahrscheinlichkeit mit den bereits erläuterten Hidden-Markov-Modellen berechnet. Dazu ist das Vorliegen der Lautschrift zu jedem Wort aus dem Vokabular notwendig (Abbildung 6.11: »Vokabular in Lautschrift«; sprachenspezifisch, aber nicht sprecherspezifisch). Zu jedem Phonem gibt es ein Markov-Modell und jedem Wort ist eine Markov-Kette zugeordnet. Die notwendige phonetische Statistik wird sprecherabhängig zum Beispiel während des Initialtrainings generiert.

Sprachmodell

Im zweiten Schritt der Erkennung wird mit Hilfe des Sprachmodells die Anzahl der Wortkandidaten auf 14 Wörter eingeschränkt, wobei die Auswahl mit Hilfe der Trigrammstatistiken, die zu jedem Wort gespeichert sind, erfolgt. Hierbei wird geprüft, welches Wort mit welchen beiden anderen in welcher Schreibweise bisher am häufigsten aufgetreten ist. Diese Trigrammstatistiken wurden aus einem extrem großen Textkorpus gewonnen, zum Beispiel wurden für den juristischen Spezialwortschatz über 500 MB ASCII-Text zur Erstellung des Wortschatzes herangezogen, das entspricht 250.000 Seiten mit je 2000 Zeichen.

Im sogenannten »Cache Language Model« befinden sich die vom Benutzer aufgenommenen Wörter mit ihren zugehörigen Trigrammstatistiken. Im Gegensatz zu den Referenzmustern, der phonetischen Statistik und der phenemic Statistik, die allesamt während des Trainings aufgebaut werden, wird das Cache Languange Model während der Arbeit mit dem System erstellt (Abbildung 6.11: »sprecherspezifische Spalte»).

Detailliertes akustisches Modell

Die Endkandidaten für die Erkennung werden im detaillierten akustischen Modell selektiert. Diese letzten drei Wortkandidaten werden der Wahrscheinlichkeit nach geordnet und das Wort an erster Stelle wird in den Text aufgenommen, nachdem es zuvor noch einmal dem Dekodierer zugeführt wurde.

Diese Selektion erfolgt technisch ähnlich wie im schnellen akustischen Modell, wobei nun nicht die Lautschrift zu jedem Wort vorliegt, sondern zu jedem Phonem eine Symbolfolge existiert. Eine Kette solcher Symbolfolgen repräsentiert ein Wort. Solche Ketten werden durch die Aussprache der Wörter durch mehrere Sprecher gewonnen. Die Enden der wahrscheinlichsten Erkennungspfade bilden zugleich die Anfangspunkte für einen erneuten Durchlauf der Ketten durch den Dekodierer.

6.6 Technische Daten zum IBM VoiceType-Diktiersystem

Wie bereits erwähnt wurde, ist das IBM VoiceType Diktiersystem eine Weiterentwicklung des TANGORA-Systems bzw. der ISSS. Über 90% des Maschinencodes des TANGORA-Systems wurden im IBM VoiceType Diktiersystem verwendet. Dadurch kann das System auf eine jahrzehntelange Geschichte der Weiterentwicklung und Verbesserung zurückblicken.

6.6.1 Verwendete Erkennungstechnik

Wie bei den meisten derzeit verfügbaren Systemen werden auch beim IBM-VoiceType-Diktiersystem die Techniken der Hidden-Markov-Modelle zur Spracherkennung verwendet. Zusätzlich jedoch wird eine weitere Möglichkeit geschaffen, die Erkennungsleistung des

Systems zu steigern. Dabei kommt die sogenannte Trigramm-Technik zum Einsatz. Trigramme sind Dreierwortverbindungen, das heißt, es werden immer drei Wörter zusammen nach der Häufigkeit ihres gemeinsamen Vorkommens statistisch erfaßt. Durch diese Kontextprüfung von jeweils drei Wörtern kann das System ähnlich oder sogar gleich klingende Wörter (Homophone) sicher unterscheiden, da es für jedes Wort seine Verwendung im Kontext und die Schreibweise der beteiligten Wörter im entsprechenden Umfeld protokolliert. Nur aufgrund dieser Technik kann dieses Diktiersystem Homophone in dieser Qualität unterscheiden. Der oben erwähnte Satz mit den Fliegen wäre bei einem anderen System nicht ohne Fehler möglich. Die Erkennungsleistung beträgt beim IBM-VoiceType-Diktiersystem ca. 70 bis 100 Wörter pro Minute, abhängig von der Artikulationsgeschwindigkeit des Sprechers.

6.6.2 Betriebssysteme

Aufgrund der extrem komplizierten und aufwendigen Berechnungen sowohl für die Hidden-Markov-Modelle, als auch für die Trigramm-Statistiken wird ein Betriebssystem benötigt, das die Leistungsfähigkeit der heutigen Prozessoren optimal nutzen kann. Deshalb wurde das IBM VoiceType Diktiersystem zunächst nur für OS/2 auf dem Markt eingeführt, denn OS/2 bietet volle Nutzung der 32-Bit-Prozessoren. Auf der CeBIT 1995 wurde zum ersten Mal das IBM VoiceType Diktiersystem für Windows 3.1 vorgestellt, das zu diesem Zeitpunkt bereits auf dem Markt verfügbar war. Das IBM-VoiceType-Diktiersystem wurde in diesem Fall als Win-32s-Applikation implementiert, um auch unter Windows so gut wie möglich Prozessoren der 486- und Pentium Familien auszunutzen. Dennoch war die Geschwindigkeit des Windows-Systems ca. 10% langsamer als die OS/2-Version. Im Jahre 1996 wurde ebenfalls auf der CeBIT VoiceType 3.0 für Windows 95 vorgestellt mit erheblich erweitertem Funktionsumfang. Da ab dieser Version keine spezielle Hardwarekarte mehr notwendig war und eine Soundblaster 16 zur Spracherkennung ausreichte, wurde der Preis von ca. 2500,- DM auf 1500,- DM reduziert. Die Technik dieser Version entspricht weitgehend derjenigen, die in OS/2 Warp 4 integriert ist.

6.6.3 Hardwarevoraussetzungen

Im Laufe der Weiterentwicklung haben sich die Anforderungen an die Hardware dramatisch verändert. Ein großer Schritt liegt vor allem in den »alten« Versionen 1.x und den derzeit verfügbaren Versionen von VoiceType, deren Technik in OS/2 Warp 4 integriert oder für Windows95 als VoiceType Version 3 verfügbar ist.

VoiceType-Versionen 1.x

Wie bereits oben angeklungen ist, benötigte das Diktiersystem in den Versionen 1.x einen Prozessor der 486-Familie ab einem 486DX25 MHz aufwärts. Zur Spracherkennung waren 8 Mbyte zusätzlich zu OS/2 beziehungsweise Windows notwendig, das heißt mit 12, besser 16 Mbyte konnte bereits optimal diktiert werden. Auf der Festplatte wurden durch das System ca. 30 Mbyte belegt. Die persönlichen Sprachdateien, die die zusätzlich aufgenommenen Wörter sowie die persönlichen Sprachmuster enthalten, umfaßten pro Benutzer ca. 1 Mbyte. Die

Größe der Sprachmuster gilt auch für die aktuellen Versionen der Spracherkennung. Da die Dateien also sehr klein sind, kann man sehr leicht seine persönlichen Sprachdateien von einem PC auf einen anderen übertragen. Dadurch wird das System sehr flexibel, weil an einem PC mehrere Benutzer registriert sein können, gleichzeitig kann sich auch die Registrierung eines Benutzers auf mehreren PCs befinden. Der Grund dafür, daß bei dem IBM VoiceType Diktiersystem Version 1.x verhältnismäßig moderate Anforderungen an den Prozessor gestellt wurden, lag darin, daß bei diesem System eine spezielle Hardwarekarte mitgeliefert wurde. Auf dieser PC-Karte mit ISA-Architektur befand sich ein speziell von IBM entwickelter Spracherkennungsprozessor, der zwischen 20 und 25 MIPS (Millionen Befehle pro Sekunde) leisten kann. Ein weiterer großer Vorteil dieser Karte war, daß bis zu 100 diktierte Wörter für die Verarbeitung in einem 2,5 Mbyte großen dynamischen RAM zwischengespeichert wurden. Dies sind durchschnittlich acht bis zehn Sätze. Beim Diktieren brauchte der Benutzer also nicht auf die Geschwindigkeit des Systems zu achten, da es kein Wort übergeht, selbst wenn der Diktierende sehr schnell spricht. Die gesamte akustische Verarbeitung wurde von der Karte übernommen, wodurch die Prozessorbelastung im PC sehr gering blieb. Diese Hardwarekarte gab es auch als MCA-Version sowie als PCMCIA-Karte, die den mobilen Einsatz in einem Notebook erlaubte. So konnte der Benutzer sogar unterwegs seine Texte mit einer automatischen Spracherkennungslösung erfassen.

Aktuelle VoiceType-Versionen

Die aktuellen Versionen von VoiceType, das heißt, Version 3.0 für Windows 95 und das in OS/2 Warp 4 integrierte VoiceType benötigen keine spezielle Hardwarekarte mehr. Lediglich eine 16-Bit-Soundkarte ist notwendig. Dies wurde durch zwei Neuerungen erreicht. Zum einen ist die Rechenleistung der modernen Prozessoren vom Typ Pentium oder PentiumPro so enorm gestiegen, daß die Rechenarbeit von diesen Prozessoren übernommen werden kann. Zum anderen wurde die Erkennungstechnik der Software so optimiert, daß nun wesentlich weniger Rechenaufwand bei ähnlicher Erkennungsleistung notwendig ist. Dennoch ist mindestens ein Pentium mit mindestens 90 MHz und 24 Mbyte Hauptspeicher Voraussetzung für akzeptabel schnelles Arbeiten mit VoiceType. An Speicherplatz auf der Festplatte werden etwa 40 Mbyte benötigt.

6.6.4 Umfang des Wortschatzes

Das IBM VoiceType Diktiersystem verwaltet je nach verwendetem Wortschatz unterschiedlich viele Wörter. Maximal kann es etwa 60.000 Wörter im aktiven Wortschatz verwalten. Wie bei allen Spracherkennungssystemen sind hier Wortformen gemeint, wenn von Wörtern die Rede ist. Zusätzlich zu den bereits im Wortschatz gespeicherten Wörtern kann der Benutzer noch eigene Wörter hinzufügen, um den Wortschatz auf seine persönlichen Bedürfnisse anzupassen und zu optimieren. Wichtig bei der Beurteilung der Wortschatzgröße ist, wie groß der aktive Wortschatz ist. Das ist die Anzahl der Wörter, die das System permanent im Speicher hält und so direkt darauf zugreifen kann. Ein Lexikon im Hintergrund der Spracherkennung hat im Prinzip keinen Wert für die Erkennungsleistung. Der einzige Vorteil besteht darin, daß nach Erkennungsfehlern die Korrektur einfacher vonstatten gehen kann (man tippt das neue Wort nicht ein, sondern wählt es aus dem Lexikon aus).

6.6.5 Verfügbare Wortschätze für Spezialisten

Prinzipiell kann ein Spracherkennungssystem um so besser ein einzelnes Wort erkennen, je eingeschränkter der Wortschatz ist. Andererseits muß ein Wortschatz über eine entsprechende Größe verfügen, damit sinnvoll gearbeitet werden kann.

Die Entwickler des IBM-VoiceType-Diktiersystems sind hier einen gewissen Kompromiß eingegangen: Statt den Wortschatz immer weiter zu vergrößern, und damit das Risiko in Kauf zu nehmen, daß die Geschwindigkeit und die Präzision geringer werden, wurden für spezielle Fachbereiche eigene Vokabulare entwickelt. Standardmäßig wird mit dem Diktiersystem der Wortschatz für die allgemeine Büroumgebung mitgeliefert. Weitere Fachvokabulare sind ebenfalls verfügbar. So zum Beispiel für Orthopädie, Radiologie, Gynäkologie, innere Medizin, Unfallchirurgie, HNO, technische Gutachten, Steuerberatung und Wirtschaftsprüfung, um nur einige zu nennen. So kann jede Person, die in einem speziellen Umfeld diktieren muß, ihren optimalen Wortschatz verwenden und damit von Anfang an auf die spezifischen Fachwörter und Ausdrucksweisen zurückgreifen, ohne daß sie selbst viel korrigieren muß. Dem Benutzer wird dadurch die Einarbeitungszeit stark verkürzt, weil er dem System keine Fachwörter beibringen muß. Für Großkunden und Anwender im großen Stil bietet sich auch die Möglichkeit, sich einen eigenen Wortschatz speziell für die eigenen Anforderungen erstellen zu lassen. So hat sich zum Beispiel das Warenversandhaus Quelle in Fürth einen eigenen Wortschatz erstellen lassen.

6.6.6 Sprechertraining zu Beginn

VoiceType verlangt vor Beginn der Arbeit kein Training mehr, wie es in früheren Versionen noch notwendig war. Um jedoch eine möglichst gute Erkennung von Anfang an gewährleisten zu können, kann das System optional vor dem eigentlichen Verwenden auf die Stimme des Sprechers trainiert werden. Dabei kann zum einen ein Training für die Navigation, also die Steuerung der Arbeitsoberfläche, durchgeführt werden, zum anderen gibt es auch ein Trainingsprogramm für die Diktierfunktion. Dabei lernt der Benutzer, wie er selbst am besten die isolierende Sprechweise verwenden kann. Während dieser Trainingsphase lernt also das System, aber auch der Benutzer, mit der neuen Art der Mensch-Maschine-Kommunikation umzugehen.

Das Training, beim IBM VoiceType Diktiersystem »Registrierung« genannt, läuft folgendermaßen ab: Das System gibt etwa 250 Sätze dem Benutzer vor. Diese muß der Diktierende Wort für Wort nachsprechen. In diesem Text wird einiges über die Hintergründe der automatischen Spracherkennung, ihre Geschichte und ihre Einsatzmöglichkeiten erzählt. Auch spezielle Fremdwörter mit Lauten, die im Deutschen nicht vorkommen, werden trainiert. Dadurch kann das System auch Lehnwörter aus anderen Sprachen gut erkennen.

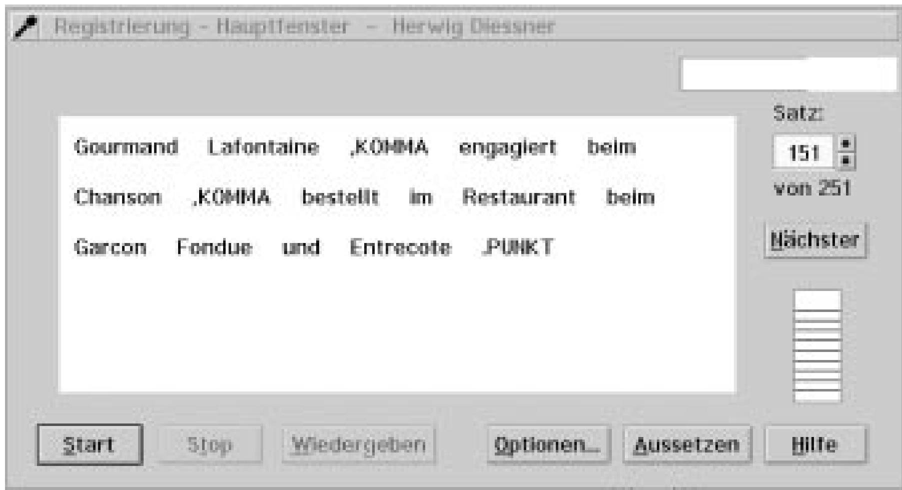


Abb. 6.13: Registrierung des IBM-VoiceType-Diktiersystems

Die Sätze sind so speziell ausgewählt, daß alle wichtigen Laute, die im Deutschen vorkommen können, vom Benutzer gesprochen werden, ohne daß er explizit darauf hingewiesen wird. Nach dem Abschluß dieser Registrierung, die etwa eine Stunde dauert, berechnet das System aus der aufgenommenen Akustik die persönlichen Sprachdateien des Benutzers. Mit diesen Dateien kann das System im Prinzip vorhersagen, wie der Benutzer alle Wörter des Wortschatzes aussprechen wird. Selbst wenn der Benutzer also ein Wort zum ersten Mal diktiert, wird es bei diesem System in der Regel sofort richtig erkannt. So wird bereits zu Beginn des Diktierens eine Erkennungsrate von etwa 95% erreichbar, wenn sich die diktierten Wörter schon im mitgelieferten Wortschatz befinden.

Das Training zur Steuerung der Oberfläche weist einige Unterschiede zum Training für das Diktat auf. So muß hier nicht mehr isoliert gesprochen werden. VoiceType kann bei der Navigation die kontinuierliche Sprechweise verstehen. Der Benutzer lernt im Laufe dieses Trainings die wichtigsten Befehle zur Steuerung und zum Buchstabieren kennen.

6.6.7 Arbeitsweise des Benutzers

Beim Diktieren mit IBM VoiceType werden Satz- und Sonderzeichen mitdiktiert. Außerdem ist es möglich, die Arbeitsoberfläche von OS/2 bzw. die Oberfläche von Windows mit Sprachbefehlen zu steuern. Es können Fenster geöffnet und geschlossen, Anwendungsprogramme gestartet, bedient und beendet werden und dies alles mit gesprochenen Befehlen. Auch das Starten des Diktats ist mit dem Befehl »Diktat starten« möglich.

Die Philosophie, die beim Arbeiten mit VoiceType zum Tragen kommt, bedeutet, daß der Diktierende während des Diktates absolut keinen Blickkontakt zum Bildschirm benötigt. Er kann also »eyes-free« diktieren und sich erst dem Computer wieder zuwenden, nachdem das Diktat beendet ist.

Durch diesen Vorteil läßt sich dieses System in der Praxis besser in den Arbeitsablauf seiner Benutzer integrieren: Ein Radiologe soll z.B. ein Röntgenbild befunden. Er steht dabei vor dem Projektionsschirm, betrachtet das Bild und möchte diktieren. Dabei hat er meistens nicht die Möglichkeit, nach jedem gesprochenen Wort sich dem Computer zuzuwenden um zu prüfen, ob er richtig erkannt wurde. Beim IBM-VoiceType-Diktiersystem kann er sein Diktat durchführen, während er konzentriert auf das Röntgenbild blickt. Anschließend kann er sich, nachdem er den Befehl »Diktat stoppen« gab, dem Computer zuwenden, den gesamten Text nachlesen, anhören und gegebenenfalls korrigieren.

Ähnlich geht es dem Juristen, der beim Studium seiner Akten Aufzeichnungen diktieren möchte. Er kann aus Gesetzestexten zitieren und sich voll auf seine Vorlagen konzentrieren. Nachdem der Text per Diktat erfaßt wurde, wird er sich konzentriert der Korrektur zuwenden.

Das ist nur möglich, weil die gesamte Akustik des Diktates auf die Festplatte des Computers aufgezeichnet wurde. Der Benutzer kann sich so den gesamten Text wie bei einem Diktiergerät wiedergeben lassen. Trat eine Fehlerkennung auf, so kann er die Wiedergabe unterbrechen, sich gezielt das einzelne Wort noch einmal akustisch wiedergeben lassen und dann prüfen, weshalb der Fehler entstanden ist. Wenn er sich versprochen haben sollte, oder ein Störgeräusch in der Umgebung die Fehlerkennung hervorgerufen hatte, kann er das Wort löschen und erneut diktieren. Falls tatsächlich das System falsch erkannt hat, kann er an dieser Stelle das Wort korrigieren. Dazu kann er seine Stimme, die Maus oder die Tastatur benutzen. Wenn er die Korrektur des Wortes beginnt, wird ihm zunächst die Akustik wiedergegeben und ähnlich klingende Wörter zur Auswahl angeboten. Befindet sich das korrekte Wort nicht in dieser Liste, so muß der Benutzer das Wort mit der Tastatur eingeben, ein Buchstabieren des Wortes ist seit der neuen Version ebenfalls möglich.

Bisher kann keine Standardanwendung unter OS/2 oder Windows die Akustik von Diktaten aufnehmen, deshalb muß der Benutzer in einem spezielles Diktierfenster seinen Text erfassen. Die Software-APIs wurden jedoch von IBM zur Verfügung gestellt, das heißt, jeder Softwareentwickler könnte seine Anwendung »speech-aware« programmieren, so daß man direkt in die Anwendung diktieren kann. Der Diktierende ist also bisher noch auf das Diktierfenster angewiesen und muß den Text nach dem Diktat und nach der Korrektur in seine Zielanwendung per Makro übergeben. Im Diktierfenster können keine Formatierungen vorgenommen werden, es handelt sich hierbei um reinen ASCII-Text, der natürlich auch als Textdatei gespeichert werden kann. Wie beim professionellen Satz erfolgt also zunächst die reine Mengentext-Erfassung, die Formatierung erfolgt erst in der Zielanwendung. Dadurch ist der Benutzer unabhängig von jeglicher Software und ihrer speziellen Bedienungsphilosophie. Es gibt nun in der neuen Version von VoiceType auch ein Fenster zum »schnellen Diktat«, in welches man diktieren kann und den Text direkt in die Zielanwendung übergeben wird. So kann man auch in Anwendungen, die nicht »speech-aware« geschrieben wurden, quasi direkt diktieren.

6.6.8 Sinn der isolierten Spracherkennung

Personen, die zum ersten Mal mit einem Spracherkennungssystem konfrontiert werden, das mit isolierter Worterkennung arbeitet, sind oft erstaunt oder gar schockiert darüber, daß man so »abgehackt« sprechen muß. Sie könnten sich absolut nicht vorstellen, auf eine ähnliche Art

und Weise zu sprechen. Sie heben in ihrer Kritik das Argument hervor, die isolierte Worterkennung setze ein unnatürliches Verhalten beim Sprechen voraus, weil man nach jedem Wort eine kleine Pause machen muß.

In der Regel spricht man mit anderen Personen in fließender Sprache, das heißt, ohne nennenswerte Pausen zwischen den einzelnen Wörtern. Dagegen muß man jedoch einwenden, daß das Trennen von Sätzen in Wörter oder kurze Phrasen in unserer Gesellschaft zu den grundlegenden Kulturtechniken gehört. Möchte man sicherstellen, daß das wörtliche Verstehen beim Gesprächspartner gewährleistet ist, so trennt man automatisch die Sätze in einzelne Wörter (zum Beispiel wenn man sich mit ausländischen Gästen unterhalten möchte). Mit dem IBM-VoiceType-Diktiersystem kann man 70 bis 100 Wörter pro Minute diktieren, das entspricht der Geschwindigkeit einer deutlich gesprochenen Rede. Dabei empfindet der Hörer es nicht unbedingt als störend, daß die einzelnen Wörter voneinander getrennt sind.

Diese Sprechweise erfordert zwar ein gewisses Training, aber bereits nach kurzer Zeit hat sich der Diktierende an diese Voraussetzung gewöhnt. Bei der Korrektur jedoch bringt die isolierte Sprechweise große Vorteile, weil Fehlerkennungen oder Sprechfehler des Benutzers leichter lokalisiert und korrigiert werden können. Tritt ein Fehler bei der kontinuierlichen Spracherkennung auf, so könnte der gesamte Satz völlig falsch erkannt werden, obwohl zum Beispiel nur ein Vokal falsch betont wurde. Dem Benutzer wäre dies völlig unverständlich und er könnte die Gründe für diese Fehlerkennung nicht nachvollziehen. Die logische Folge daraus wäre, daß der Benutzer frustriert wird. Bei der isolierten Spracherkennung ist es dem Benutzer jedoch möglich, das System als mächtiges und brauchbares Werkzeug für die Mensch-Maschine-Kommunikation zu verwenden, die Spracherkennung ist dabei nicht Ersatz für Tastatur und Maus, sondern ein drittes, ergänzendes Eingabemedium.

6.7 Workflow bei der Dokumentenerstellung

In diesem Kapitel soll untersucht werden, wie sich der Arbeitsablauf in einer Kanzlei oder im Büro durch den Einsatz von automatischen Spracherkennungssystemen auf dem PC verändert. Zunächst wird der bisherige Arbeitsfluß untersucht und danach die neue, durch computergestützte Spracherkennung veränderte Arbeitsweise betrachtet.

6.7.1 Bisheriger Arbeitsablauf beim Erstellen von Dokumenten

Der konventionelle Arbeitsablauf beim Erstellen eines Textes, der zuvor diktiert wurde, wird durch Abbildung 6.14 veranschaulicht.

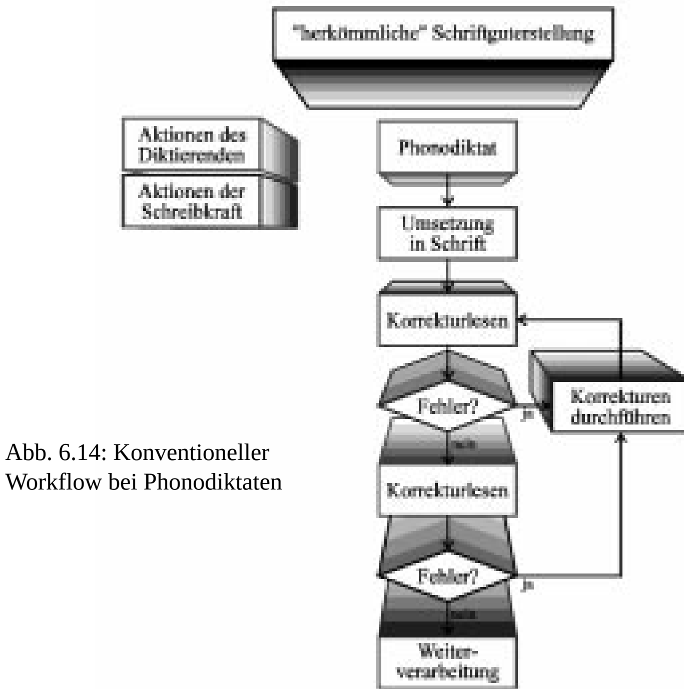


Abb. 6.14: Konventioneller Workflow bei Phonodiktaten

Für die vollständige Bearbeitung eines diktierten Dokuments sind mindestens zwei Personen notwendig: Zum einen der Diktierende zum anderen die Person, die das Dokument bearbeitet.

Phonodiktat

Die Person, die einen Text erfassen möchte, diktiert ihre Gedanken auf ein konventionelles Diktiergerät, also auf Band. Dabei muß sie sich im Prinzip an keine Konventionen für das Diktieren halten. Ist das Diktat beendet, gibt der Diktierende das Band an eine andere Person zur weiteren Bearbeitung ab. Die einzige Voraussetzung dazu ist, daß die andere Person das Band abhören und den Inhalt des Diktates verstehen kann.

Umsetzung in Schrift

Diese Bearbeitung wird in der Regel von einer Schreibkraft im Schreibbüro oder von einer Sekretärin erledigt. Sie muß das Band abhören und das Diktat in Text umsetzen. Dabei kann sie auch gleich die Formatierung des Textes durchführen. Da jedoch eine Sekretärin oft auch andere Aufgaben zu erledigen hat oder das Schreibbüro oft überlastet ist, dauert die Umsetzung in Schrift in der Regel längere Zeit.

Korrekturlesen (Schreibkraft)

Ist die Texterfassung abgeschlossen, so wird der Text auf orthographische Fehler hin untersucht, bevor das Dokument an den Diktierenden zurückgeleitet wird. Dies übernimmt die Schreibkraft oder die Sekretärin, die auch das Phonodiktat in Text umgewandelt hat. Falls Tippfehler aufgetreten sind, muß die Schreibkraft oder die Sekretärin den Text in vollem Umfang erneut schreiben oder, falls ein Computer eingesetzt wird, noch einmal drucken.

Korrekturlesen (Diktierender)

Das vorläufige Dokument wird an den Diktierenden zurückgegeben, damit diese Person Korrektur lesen kann. Dabei überprüft sie hauptsächlich inhaltliche Aspekte, ob im Text inhaltliche Fehler enthalten sind, ob der Text vollständig ist oder überflüssige Teile enthalten sind. Falls derartige Fehler tatsächlich aufgetreten sind, muß der Text erneut vom Schreibbüro bzw. von der Sekretärin bearbeitet und ganz neu geschrieben werden. Ist dies geschehen, wird der Text, nachdem die Schreibkraft den Text auf orthographische Fehler hin überprüft hat, erneut an den Diktierenden zur Unterschrift zurückgegeben. Dieser Vorgang wiederholt sich so lange, bis ein fehlerfreies, vollständiges Dokument erstellt ist, das nun zur Weiterverarbeitung freigegeben werden kann.

Weiterverarbeitung

Dies könnte zum Beispiel der Versand an Dritte oder die Ablage des Textes sein. Oft müssen die Texte auch kopiert werden oder als Hausmitteilungen im Unternehmen verteilt werden.

6.7.2 Neuer Workflow beim Einsatz von Diktiersystemen

Der neue Workflow, der durch automatische Spracherkennungssysteme, die auf dem PC basieren, erreicht werden kann, wird in der folgenden Abbildung 6.15 aufgezeigt.

PC-Diktat

Der Diktierende sitzt an seinem PC und dieses Gerät erfaßt bereits während des Diktates den Rohtext des Dokuments. Dabei gelten die bereits beschriebenen Bedingungen für ein PC-Diktat. Da der Diktierende in isolierender Sprechweise diktieren muß, wird er zunächst beim Diktieren etwas langsamer werden, da er sich erst an die neue Sprechweise gewöhnen muß. Hat er sich jedoch erst einmal umgestellt, so wird ihm auch diese Art und Weise zu diktieren leicht von der »Zunge« gehen. Mit den heutigen Spracherkennungssystemen ist es möglich, 70 bis 100 Wörter pro Minute diktieren zu können. Nur wenige gute Sekretärinnen können solche Werte bei der Textumsetzung erreichen.



Abb. 6.15: Optimierter Workflow durch Spracherkennungssysteme

Sofortige Korrektur

Noch während sich der Text auf dem Bildschirm des PCs befindet, wird der Text korrigiert. Das heißt also, der Diktierende korrigiert seinen Text sofort oder im Anschluß an das Diktat eigenhändig. Einige Personen meinen zunächst, dadurch würden sie in ihrer Produktivität stark eingeschränkt werden, denn sie müssen wertvolle Zeit für die Korrektur aufwenden. Dagegen ist jedoch folgendes Beispiel einzuwenden. Ein Radiologe einer größeren Klinik, der vor kurzem ein IBM VoiceType Diktiersystem bestellte, beschrieb seine bisherige Situation folgendermaßen:

Bevor ein von ihm erstelltes Phonodiktat im Schreibbüro bearbeitet werden konnte und ihm zur Korrektur zurückgebracht wurde, vergingen oft bis zu vier Wochen. Nach diesem Zeitraum ist ihm natürlich nicht sofort gegenwärtig, was er genau bei diesem Patienten diagnostiziert hatte. Er mußte sich den Befund wieder vorlegen lassen und durchlesen, um sich erneut in den Fall einzuarbeiten, wobei das Risiko eines Irrtums möglicherweise ansteigen könnte. Die Zeit, die er für diese Aktion benötigte, war nach seinen Aussagen wesentlich länger, als das sofortige eigenhändige Korrigieren des Diktates.

Die neuen Versionen von VoiceType bieten jedoch auch die Möglichkeit eine Diktiersitzung abzuspeichern, das heißt die Akustik und den erkannten Text für eine spätere Korrektur aufzubewahren. Diese Korrektur könnte dann auch eine andere Person übernehmen. Bei der in OS/2 Warp 4 integrierten Version von VoiceType steht diese Funktion erst dann zur Verfügung, wenn das zusätzlich erhältlich Upgrade oder ein Spezialwortschatz installiert wurden.

Ein Vorteil des PC-Diktates ist außerdem, daß in der Orthographie keine Fehler auftreten. Die Wörter, auch wenn es sich um komplizierte Fachwörter handelt, sind im Vokabular orthographisch richtig enthalten, die einzige mögliche Fehlerquelle ist eine falsche Erkennung des gesamten Wortes. Beim Korrekturlesen fällt dies jedoch sofort auf, man kann also die Fehler leichter erkennen, denn es brauchen nicht die einzelnen Buchstaben kontrolliert werden.

Weiterverarbeitung

Nachdem der Text diktiert und korrigiert wurde, kann er gespeichert werden und von der Sekretärin bzw. dem Schreibbüro weiterbearbeitet werden. Da beim IBM-VoiceType-Diktiersystem keine Formatierungen mitdiktiert werden, kann erst zu diesem Zeitpunkt die Formatierung des Textes durchgeführt werden. Dadurch wird die Sekretärin bzw. das Schreibbüro stark entlastet, weil das Anhören des Diktates entfällt. Das Formatieren des Textes wird aber weiterhin von der Sekretärin übernommen, ebenso die weitere Verarbeitung bzw. Verwendung des Dokuments. Die Umsetzung der Akustik in Text wurde automatisch durchgeführt und dadurch ist es möglich, daß zum Beispiel die Sekretärin in der »gesparten« Zeit andere, interessantere und meist höherwertige Tätigkeiten durchführen kann.

6.7.4 Zusammenfassung der Vorteile

Durch den Einsatz von PC-gestützten Spracherkennungssystemen können folgende Vorteile erzielt werden: Nach einer kurzen Einarbeitungszeit des Diktierenden, meist etwa zwei bis vier Wochen, können Diktate beinahe genauso schnell erfaßt werden, wie beim herkömmlichen Phonodiktat. Dabei profitiert man aber von dem Vorteil, daß der Text sofort auf dem Bildschirm erscheint und, sobald das Diktat beendet wurde, der Rohtext zur Korrektur vorliegt. Weil die Korrektur sofort nach dem Diktat erfolgt, entfallen für den Diktierenden spätere Einarbeitungszeiten, der Diktierende ist noch mit der Materie vertraut.

Aber nicht nur von seiten des Diktierenden werden Vorteile erlangt. Auch die Personen, die bisher den Text erfaßt und weiterbearbeitet haben, werden durch PC-Diktate in ihrer Arbeit entlastet. Wer schon einmal mit Büroarbeit in Berührung kam, wird sicher auch die Erfahrung gemacht haben, daß das häufige Wiederholen von Schreiben und Korrigieren eines Dokuments die Nerven strapazieren und Streß erzeugen kann. Hauptsächlich wird die Durchlaufzeit für ein Dokument durch Spracherkennung enorm optimiert. Das gesamte Team wird also wesentlich produktiver und streßfreier als bei dem herkömmlichen Vorgehen mit Phonodiktat und nachträglichem Umsetzen in Schrift arbeiten können.

6.7.5 Zusammenfassung der Nachteile

Bisher waren vor allem die hohen Anschaffungskosten eine große Hemmschwelle, sich für ein Spracherkennungssystem zu entscheiden. Bis vor kurzem konnte eine praktikable Hard- und Softwarelösung leicht die 50.000-DM-Grenze übersteigen. Seit jedoch zuverlässige Systeme auch für Standard-PCs angeboten werden, sind die Kosten kaum noch ein Argument. In der Regel hat sich solch ein System bereits innerhalb der ersten Monate des Einsatzes amortisiert.

Ein größeres Problem ist die Disziplin, die ein solches System von seinen Benutzern fordert. Es muß deutlich und mit Pausen gesprochen werden und die Korrektur muß gewissenhaft durchgeführt werden. Wird die Korrektur vernachlässigt, sinkt die Erkennungsleistung immer mehr ab, wobei der Benutzer diese Tatsache meist als Unzulänglichkeit des Systems interpretiert.

Die größte Gefahr im Einsatz solcher Systeme wird darin gesehen, daß durch spracherkennende PCs die Arbeitsplätze der Schreibkräfte in Gefahr kommen. Fakt ist jedoch, daß man auch weiterhin auf das Sprachverständis einer Schreibkraft nicht verzichten kann, ein Büro ganz ohne Schreibkräfte wird es wohl nicht geben. Wohl ist es jedoch denkbar, daß die Anzahl der Schreibkräfte verringert werden kann, da diesen zum Teil die Arbeit abgenommen wird. Positiv dabei ist jedoch die Tatsache, daß dadurch das Unternehmen kosteneffektiver arbeiten kann.

6.8 Aussichten für die Zukunft

Bereits heute können auf dem PC basierende Diktiersysteme in vielen Bereichen effektiv eingesetzt werden, wobei die Arbeit folgenden Beschränkungen unterliegt:

– In einigen Sprachen reicht der Umfang des Wortschatzes nicht aus

– Pausen nach den Wörtern, damit die Erkennungsgeschwindigkeit steigt

Es gibt viele Spekulationen und Vermutungen, wie Spracherkennungssysteme in der Zukunft implementiert und realisiert werden könnten. Das große Ziel wird natürlich sein, die oben genannten Beschränkungen zu beseitigen. Bisher sind jedoch solche Systeme, wenn überhaupt, nur als Labormuster und Prototypen und hinter verschlossenen Türen zu finden. Durch die fortschreitende Miniaturisierung der Hardware wäre auch die zukünftige Integration der Spracherkennungstechnologien in sogenannte PDAs (Personal Digital Assistants), also tragbare Kleinstcomputer, denkbar.

Das Sprachverstehen wird in Zukunft ebenfalls eine wichtigere Rolle spielen. Kann ein Computer Sprache wirklich verstehen, so sind Dialoge zwischen Mensch und Computer möglich. Denkbare Anwendungen wären zum Beispiel der elektronische »Kiosk« oder die automatische Übersetzung in beliebige Sprachen.

Bei all den zukünftigen Entwicklungen, wie sie auch immer sein mögen, sollte stets das oberste Ziel dabei sein, daß der Mensch und nicht die Maschine im Mittelpunkt steht (human centered technologies).

Die Einführung der Spracherkennung ist in gewisser Weise mit der Integration der Maus vergleichbar: Als die ersten Computer mit einer Maus ausgerüstet wurden, gab es kaum Software, die durch eine Maus bedient oder gesteuert werden konnten. Auch die Integration in das Betriebssystem verursachte zunächst Probleme. Heute ist praktisch jede Anwendung mit Maus zu bedienen und in den modernen Betriebssystemen mit grafischer Oberfläche wird die Maus standardmäßig voll unterstützt. In einer ähnlichen Situation befindet sich derzeit die Spracherkennung auf dem PC. Zunächst noch skeptisch betrachtet und kaum in eine Anwendung integriert, wird in Zukunft die Bedeutung der Spracherkennung stark zunehmen und bald nicht mehr aus dem täglichen Arbeiten mit dem Computer wegzudenken sein. Mit OS/2 Warp 4 wurde nun zum ersten Mal die Spracherkennung in das Betriebssystem integriert. Dies ist ein weiterer Schritt zur weiten Verbreitung und Akzeptanz der neuen Technologie.

6.9 Die Praxis unter OS/2 Warp 4

Erstmals wurde eine Spracherkennung in ein Betriebssystem integriert. Nun kann unter OS/2 Warp 4 die vollständige Funktionalität von VoiceType verwendet werden.

IBM VoiceType für OS/2 besteht aus zwei Komponenten:

– Der Navigation, das heißt die Steuerung der Arbeitsoberfläche durch gesprochene Kommandos, das Öffnen und Schließen von Fenstern und Anwendungen und das Arbeiten mit Anwendungsprogrammen.

– Der Texterfassung durch Diktieren, das heißt Briefe und andere Dokumente können nun ohne Eintippen in den Computer erfaßt werden, einfach durch Ihr gesprochenes Wort. VoiceType kann 70 bis 100 Wörter pro Minute in Text umwandeln.

Der VoiceType Order enthält folgende Objekte:



Im folgenden werden alle Objekte dieses Ordners der Reihe nach behandelt, nachdem die Installationsmöglichkeiten von VoiceType unter OS/2 Warp 4 erläutert wurden. Außerdem kommen auch andere Objekte der Spracherkennung zur Sprache, die in anderen Ordnern der Arbeitsoberfläche zu finden sind.

6.9.1 Installation von VoiceType

VoiceType kann auf zwei verschiedene Arten installiert werden. Entweder erfolgt die Installation zusammen mit dem Betriebssystem OS/2 Warp 4 oder nachträglich über die Option *Installation anpassen*. Zuvor jedoch noch ein Blick auf die Voraussetzungen für die Installation.

Voraussetzungen für die Installation

Bevor Sie mit der Installation von VoiceType beginnen, sollten Sie klären, ob Ihr PC die Anforderungen für einen sinnvollen Einsatz erfüllt. Auch wenn diese Anforderungen nicht erfüllt werden, so können Sie doch VoiceType installieren, allerdings wird kaum ein sinnvoller Einsatz möglich sein, da der PC in diesem Fall viel zu lange rechnet, bevor er irgendwelche gesprochenen Befehle abarbeiten kann. Sie werden also wesentlich schneller sein, wenn Sie wie gewohnt Tastatur und Maus verwenden und auf die Spracheingabe verzichten.

Zunächst benötigen Sie einen Computer, der über einen Pentium mit mindestens 90 MHz sowie 24 MB Arbeitsspeicher verfügt. Auf der Festplatte benötigen Sie etwa 30 MB freien Speicherplatz, um die Software zu installieren. Diese Angaben sind jedoch Mindestanforderungen. Vor allem durch Erweitern des Arbeitsspeichers auf zum Beispiel 32 oder 48 Mbyte können Sie enorme Geschwindigkeitssteigerungen erwarten. Beachten Sie auch, daß die akustischen Informationen, die während des Diktierens anfallen, auf der Festplatte zwischengespeichert werden und daß deshalb zusätzlicher Platz auf der Festplatte benötigt wird!

Darüberhinaus benötigen Sie ein Mikrofon und Lautsprecher sowie eine von OS/2 Warp 4 unterstützte Soundkarte. Bisher wurden folgende Soundkarten für das Arbeiten mit VoiceType von IBM getestet, es sollen jedoch sämtliche Soundkarten, die von OS/2 unterstützt werden, funktionieren:

- Compaq Audio Business (AD1848)
- CS4231/4248
- CS4232/4236
- ESS 688
- ESS 1688
- ESS 1688 in IBM ThinkPads
- ESS 1788
- ESS 1868 PnP
- ESS 1888
- IBM M-ACPA
- Jazz 16
- Mwave in Audiovation audiocard
- Mwave in IBM ThinkPads
- ProAudio Spectrum
- SoundBlaster 16
- SoundBlaster 32

– SoundBlaster AWE 32

– SoundBlaster 16 PnP

– SoundBlaster AWE 32 PnP

– Vibra 16

Informationen zu verschiedenen Mikrofonen und deren Verwendung unter VoiceType finden Sie im Kapitel *Mikrofone für VoiceType*.

Installation zusammen mit OS/2 Warp

Wenn Sie VoiceType sofort bei Ihrer Arbeit mit OS/2 an nutzen wollen, so ist es für Sie am einfachsten, die Spracherkennung mit dem Betriebssystem zusammen zu installieren. Dazu wählen Sie im Installationsbildschirm *Konfiguration und Installation von OS/2 Warp Version 4* die Option *Wahlfreie Systemkomponenten* und wählen dort die Checkbox für die Spracheingabe aus. Damit wird VoiceType standardmäßig in das Unterverzeichnis x:\VT\ installiert, wobei *x* für das Installationsverzeichnis steht.

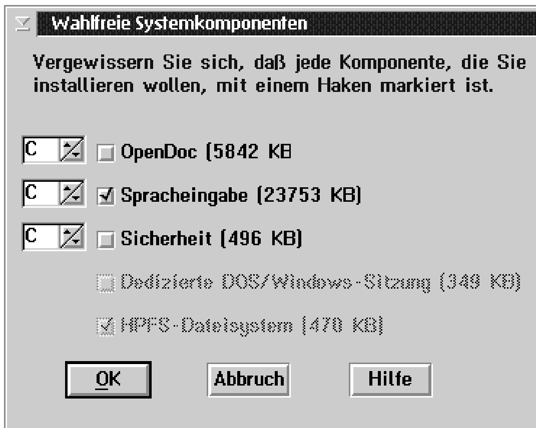


Abb. 6.16: Die Installation von VoiceType

Installation nachträglich anpassen

Entschließen Sie sich, nachträglich VoiceType zu installieren, nachdem Sie bereits eine Zeitlang mit OS/2 Warp 4 gearbeitet haben, so rufen Sie das Programm *Installation anpassen* auf. Sie erhalten denselben Installationsbildschirm wie bei der ersten Installation von OS/2 und können nun unter *Wahlfreie Systemkomponenten* die Spracherkennung auswählen und installieren.

Mikrofone für VoiceType

Ihr Computer kann nur die Daten verarbeiten, die durch das Mikrofon an ihn gelangen. Wenn die Sprachinformationen schon vom Mikrofon her sehr schlecht sind, kann er nichts erkennen!

Aus diesem Grund sind Kopfbügelmikrofone prinzipbedingt im Vorteil gegenüber Hand- oder Tischmikrofonen. Während sich die Position des Sprechers zum Mikrofon bei Hand- und Tischmikrofonen ständig ändert, sitzt das Kopfbügelmikrofon relativ zum Mund gesehen immer an derselben Position. Das Handmikrofon ist zusätzlich noch anfällig gegen Störgeräusche, die durch Handbewegungen entstehen können.

Bevor Sie mit der Spracheingabe anfangen, überprüfen Sie nochmal alles:

- Anschluß am Mikrofon-Eingang der Soundkarte (das hat schon oft alle Probleme mit einem Schlag beseitigt).
- Den Lautsprecher schließen Sie am Ausgang der Soundkarte an. *Line Out* für Lautsprecher mit Verstärker und *Speaker Out* für Lautsprecher ohne integriertem Verstärker.

Wenn VoiceType noch nicht installiert ist, können Sie folgende Punkte testen:

- SoundRecorder von OS/2 (hat nichts mit VoiceType zu tun!) öffnen, als Eingang das Mikrofon wählen, und einfach mal ein paar Wörter aufnehmen.
- Diese Aufnahme wiedergeben lassen und überprüfen, ob alles klar und deutlich wiedergegeben wird.

Meine Erfahrungen mit diversen Mikros:

(Diese Ergebnisse habe ich mit meinem PC evaluiert und ich übernehme keine Garantie, daß irgendjemand mit einem der hier erwähnten Mikrofone ähnliche Ergebnisse feststellen kann! Mein PC enthält einen Pentium mit 100 MHz und eine Soundblaster-16-Soundkarte.)

– Shure SM10A (Kopfbügelmikrofon)

Dieses Mikro wurde bisher mit früheren Versionen von VoiceType gebundlet, hat bei mir jedoch mit meiner Soundblaster 16 extreme Störgeräusche verursacht. Beim Bewegen des Mauszeigers krachte es laut im Lautsprecher. Die Lautstärke der Aufnahme war außerdem sehr niedrig.

– Mikrofon im Thinkpad 760C/CD/E/ED eingebaut

Relativ gute Erkennung, wobei ein Problem die sich ständig ändernde Position des Sprechers zum Mikrofon darstellt. Durch die kleine Mikrofon-Membran wird die Sprache nur schwach aufgenommen, Umgebungsgeräusche stören stark.

Meiner Meinung nach nur als Notbehelf zu verwenden.

– ANDREA Anti-Noise ANC-500 (Kopfbügel mit 1 Ohrhörer)

Dieses Mikrofon wird mit VoiceType Version 3.0 für Windows95 ausgeliefert. Ohrhörer und Mikrofon sind in einem Headset integriert. On/Mute Schalter. Störgeräusche werden gefiltert. Wahlweise ist eine Signalverstärkung möglich, d.h., man kann das Mikrofon perfekt an die verwendete Soundkarte anpassen. Super Aufnahmequalität!

Achtung: Wenn nicht diktiert wird, sollten alle Stecker aus dem batteriebetriebenen (2 AAA Zellen) Verstärkerteil entfernt werden, sonst sind die Batterien nach ca. 2 Monaten leer (auch ohne Diktieren).

Infos zu Andrea-Mikrofonen können Sie auf folgender Web-Seite bekommen: http://www.andreaelectronics.com

– ANDREA Anti-Noise NC-50 (Kopfbügel mit 1 Ohrhörer)

Dieses Mikrofon wird mit OS/2 Warp 4 ausgeliefert. Ohrhörer und Mikrofon sind in einem Headset integriert. Auch dieses Mikrofon filtert Störgeräusche, allerdings nicht als aktives System, sondern über eine besondere passive Schaltung des Mikrofons. Vorteil dabei ist, daß das Mikrofon robuster ist und dennoch ähnlich gute Ergebnisse liefert.

– Typhoon Sound System »CD 87 M« Kopfbügelmikrofon mit 2 Ohrhörern

Ein Kollege hatte mir dieses Mikrofon ausgeliehen, damit ich ein paar Tests zur Spracherkennung machen konnte. Die Ergebnisse waren überwältigend! Das Mikrofon kostet ca. 15,- DM und hat eine wirklich gute Erkennungsleistung vorgewiesen. Das Signal kam so stark an, daß ich an meiner SB16 den Mikrofon- Eingangspegel stark absenken konnte und somit Hintergrundgeräusche oder ein Rauschen praktisch unhörbar wurden.

6.9.2 Interaktives Lernprogramm

Das Objekt für ein interaktives Lernprogramm befindet sich nicht direkt im VoiceType-Ordner, sondern im Ordner *Unterstützung*. Durch einen Doppelklick auf *OS/2 Warp Lernprogramm* kann das Lernprogramm gestartet und das Kapitel VoiceType ausgewählt werden, in dem die folgenden Themen behandelt werden:

– Grundlagen:

Die Bestandteile von VoiceType werden ebenso beschrieben wie die zugrunde liegenden Konzepte für die Spracherkennung.

– Navigieren:

Wie kann mit gesprochenen Befehlen die Arbeitsoberfläche gesteuert werden? Welche Möglichkeiten gibt es, in Programmen Menüpunkte und Schaltflächen per Sprache zu bedienen?

– Diktieren:

Eine einführende Beschreibung der Diktierfunktionen und der Korrektur.

– Spracherkennung:

Wie kann ein Wort trainiert werden, wie erfolgt die Registrierung für die Diktier- und Navigationsfunktion, wenn man die Erkennungsgenauigkeit erhöhen will? Wie kann der Mikrofonpegel richtig eingestellt werden?

6.9.3 Installation überprüfen

Mit dem Programm *Installation prüfen* steht eine Möglichkeit zur Verfügung, die Installation des Diktiersystems umfassend auf Funktionsfähigkeit zu überprüfen. Es empfiehlt sich, dieses Programm im Anschluß an die Installation von VoiceType auszuführen, um zu testen, ob Mikrofon und Lautsprecher richtig am PC angeschlossen und funktionsfähig sind.

Nach einem Doppelklick auf das Symbol *Installation prüfen* wird folgendes Fenster angezeigt:



Abb. 6.17: Installation prüfen

Lautsprechertest

Um den Lautsprecher zu überprüfen, klicken Sie auf die Schaltfläche *Lautsprecher testen*. Wenn nichts zu hören ist, sollten Sie folgendermaßen vorgehen:

– Stellen Sie sicher, daß der Lautsprecher richtig angeschlossen ist.

– Wenn der Lautsprecher einen Lautstärkeregler besitzt, erhöhen Sie die Lautstärke.

Mikrofon testen und Eingangsspannung einstellen

Zunächst müssen Sie sicherstellen, daß sich das Mikrofon in der richtigen Position befindet. Das mitgelieferte Kopfmikrofon plazieren Sie am besten etwas außerhalb Ihres linken Mundwinkels. Die Sprechkapsel sollte sich auf keinen Fall direkt vor dem Mund befinden, denn der Luftstrom würde starke Störgeräusche beim Sprechen hervorrufen.

Daraufhin sollten Sie testen, ob das Tonsignal richtig aufgezeichnet wird, wenn Sie sprechen. Im Bereich *Mikrofon testen* ist ein kleines Tonband nachgebildet, mit dem Sie etwa 10 Sekunden lange Nachrichten aufzeichen können. Dazu klicken Sie auf *Aufzeichnen* und sprechen

einen Satz in das Mikrofon. Nach einem Klick auf *Stop* wird das nachgebildete Aufnahmeband zurückgespult und Sie können mit *Wiedergeben* den aufgezeichneten Satz anhören.

Beachten Sie während der Aufnahme die Aussteuerungsanzeige:

– Der grüne Bereich zeigt korrekte Lautstärke an.

– Gelb zeigt an, daß die Aufnahme zu leise ist. Erhöhen Sie die Aussteuerung, sprechen Sie lauter oder verkürzen Sie den Abstand zum Mikrofon.

– Eine zu laute Aufzeichnung wird durch den roten Bereich markiert. Verringern Sie die Aussteuerung, sprechen Sie leiser oder vergrößern Sie den Abstand zum Mikrofon.

– Wenn Sie gar keine Farbe in der Aussteuerungsanzeige sehen, während Sie sprechen, überprüfen Sie bitte die Verbindung des Mikrofons. Falls das Mikrofon Batterien benötigt, stellen Sie sicher, daß diese aufgeladen sind und richtig in das Batteriefach eingesetzt sind. Verfügt das Mikrofon über einen Ein-Aus-Schalter, schalten Sie das Mikrofon an.

Das Justieren des Eingangspegels können Sie auch VoiceType selbst überlassen. Markieren Sie einfach das Kästchen *Eingangsspannung auto. einstellen* und nehmen Sie danach einfach ein paar (mindestens acht) Wörter auf. Achten Sie dabei darauf, daß Sie in genau der Lautstärke sprechen, die Sie auch beim Diktieren verwenden werden.

Erkennungstest bei laufender Spracheingabe

Im Bereich *Erkennungstest bei laufender Spracheingabe* können Sie ein paar Kommandos testen und je nachdem, wie gut Sie VoiceType erkennt, die Erkennungsgenauigkeit und den Aufnahmepegel mit jeweils einem Schieberegler anpassen. Es ist empfehlenswert, den Regler für die Erkennungsgenauigkeit möglichst nicht zu verändern. Nur im äußersten Notfall, wenn VoiceType in einem anderen Fall wirklich nichts erkennt, sollten Sie hier vorsichtig und in kleinen Schritten Änderungen vornehmen. Wenn Sie eine Änderung nach einer Registrierung vornehmen, kann es sein, daß Sie noch einmal das gesamte Training durchühren müssen, da Sie grundlegende Parameter der Spracherkennung verändert haben.

6.9.4 VoiceType-Spiel

Das Spracheingabespiel soll Personen, die noch nie mit einer Spracherkennung gearbeitet hatten, den Einstieg auf spielerische Weise erleichtern. Nach dem Aufruf des Länderspiels wird eine Landkarte von Europa angezeigt. Der Benutzer spricht nun nacheinander die Namen verschiedener Staaten und diese werden dann farblich hinterlegt.

6.9.5 VoiceType-Steuerzentrum

Mit einem Doppelklick auf das Symbol *Sprachsteuerung* wird das eigentliche VoiceType Programm gestartet. Dabei werden das Fenster *Sprachsteuerung* und das Fenster für die *sprechbaren Befehle* angezeigt. Diese beiden Fenster haben eine wichtige Bedeutung und überlappen deshalb alle anderen Fenster auf der Arbeitsoberfläche. Wenn man weiß, wie man mit diesen

Fenstern umgehen muß, kann die effiziente Nutzung von VoiceType enorm erleichtert werden. Im folgenden werden diese beiden Fenster etwas genauer betrachtet.

Sprachsteuerung



Abb. 6.18:
Die Sprachsteuerung

Die Sprachsteuerung umfaßt sechs Schaltflächen, die Aussteuerungsanzeige sowie eine Statuszeile, die angibt, in welchem Zustand sich VoiceType befindet und welche Aktionen des Benutzers in diesem Zustand möglich sind.

Die Aussteuerungsanzeige ist in drei Bereiche gegliedert. Links der gelbe Bereich, in der Mitte grün und rechts rot. Während der Spracheingabe sollte sich die Aussteuerung immer im grünen Bereich bewegen. Wenn die Anzeige in den gelben oder roten Bereich gelangt, sollten Sie die Aussteuerung Ihres Mikrofons mit Hilfe der *Einstellungen* entsprechend anpassen.

Die sechs Schaltflächen haben folgende Funktionen:

Mikrofontaste

Durch Klicken mit der Maus können Sie das Mikrofon ein- oder ausschalten. Wenn Sie längere Zeit keine Kommandos per Sprache geben wollen, empfiehlt es sich, das Mikrofon auszuschalten. Je nachdem, ob das Mikrofon ein- oder ausgeschaltet ist oder sich im Ruhezustand befindet, ändert diese Schaltfläche ihr Aussehen. Sie können somit immer optisch kontrollieren, in welchem Zustand sich Ihr VoiceType befindet.

Eine weitere Möglichkeit, das Mikrofon ein- oder auszuschalten ist folgende: Tippen Sie zweimal kurz die Umschalt-Taste (Shift) an. Beachten Sie dabei, daß Sie den Ruhezustand nur per Sprache aktivieren können, die Schaltfläche dient nur dazu, das Mikrofon ganz aus- oder einzuschalten.

Sprechbare Befehle

Wenn Sie auf diese Schaltfläche klicken, wird das Fenster *Sprechbare Befehle* geöffnet. Dieses Fenster und seine Funktionen werden im nächsten Abschnitt ausführlicher behandelt. Sie können das Fenster ebenso wie alle nachfolgenden auch durch Spracheingabe aufrufen.

Sprechbare Titel

Wenn Sie diesen Schalter betätigen, wird ein Fenster angezeigt, in dem alle Objekte enthalten sind, die Sie direkt per Sprache starten können. Auch dieses Fenster wird später genauer betrachtet.

Diktierfenster

Das Diktierfenster wird geöffnet, wenn Sie die vierte Schaltfläche anklicken. Dabei handelt es sich um einen Editor, dessen Besonderheit darin liegt, daß er neben typischen Standardfunktionen für einen Editor die direkte Eingabe von Texten per Diktat zuläßt. Sie können hierin Texte mit Ihrer Sprache erfassen, diese Texte korrigieren und entweder abspeichern oder in andere Programme übertragen.

Einstellungen

Mit Hilfe der Einstellungen können Sie VoiceType an Ihre gegebene Hardware und Ihren persönlichen Geschmack anpassen. So können Sie das Aussehen des Fensters *Sprachsteuerung* an Ihre Vorstellungen anpassen oder den Aufnahme- und Wiedergabepegel für Mikrofon und Lautsprecher korrekt einstellen. Auch die Genauigkeit, mit der VoiceType gesprochene Wörter analysieren soll, können Sie konfigurieren.

Hilfe

Dieser Schalter ruft die Online-Hilfe für VoiceType auf. Ob Sie ein akutes Problem mit VoiceType haben oder einfach neues und informatives lesen wollen, es lohnt sich immer, einen Blick in diese Hilfe-Funktion zu werfen.

Sprechbare Befehle

Dieses Fenster wird angezeigt, sobald Sie VoiceType starten. Es enthält zwei »Seiten«, die durch zwei Tabs markiert sind. So können Sie wählen, ob Sie die immer sprechbaren Befehle oder die für das gerade aktive Programm gültigen Befehle anzeigen lassen wollen. Dieses Fenster dient für Sie als ständige Referenz der möglichen Kommandos, die Sie per Sprache geben können. Wenn ein Wort in diesem Fenster durchgestrichen erscheint, so bedeutet dies, daß es zwar ein mögliches Kommando darstellt aber VoiceType keine Aussprache dazu vorliegen hat. Sie können diesen Befehl aktivieren, indem Sie die entsprechende Zeile markieren und auf die Schaltfläche *Trainieren* klicken. Daraufhin haben Sie die Möglichkeit, das Wort oder den Befehl für VoiceType vorzusprechen. Sobald dies geschehen ist, wird das Kommando als sprechbar angezeigt (also nicht mehr durchgestrichen).

Sprechbare Titel

Nach dem Aufruf dieses Fensters, sei es per Maus oder durch ein Sprachkommando, werden Ihnen alle Objekte angezeigt, die Sie direkt per Sprache aufrufen können. Sie lesen dazu einfach den Programmnamen und setzen den Befehl *Sprung zu* davor, also zum Beispiel »Sprung zu OS/2 System Editor«. VoiceType führt daraufhin das angegebene Programm aus.

Damit ein Programm oder Objekt Ihrer Wahl in dieses Fenster aufgenommen wird, müssen Sie dessen Einstellungen öffnen. Unter dem Punkt *Spracheingabe* können Sie dann den Befehl eingeben, mit dem Sie das Programm starten möchten. Für diesen Befehl müssen Sie noch die Akustik aufnehmen, wenn VoiceType die einzelnen Wörter des Befehls nicht kennt. Ab diesem Zeitpunkt wird dann dieses Objekt im *Sprechbare Titel* – Fenster immer angezeigt.

6.9.6 Registrierung

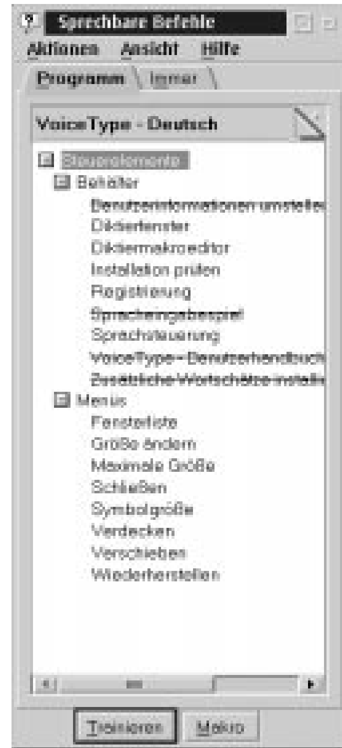


Abb. 6.19: Start von VoiceType

Mit dem Objekt *Registrierung* können Sie ein optionales Training für die Navigation oder für das Diktieren mit VoiceType durchführen Es gibt sowohl ein Training für die Navigation als auch eines für das Diktat. Dazu werden dem Benutzer einige Sätze bzw. Befehle vorgegeben, die nachgesprochen werden müssen. Aus diesen Daten kann VoiceType Ihre persönlichen Eigenheiten der Sprache und Stimme ermitteln und daraus ein Sprachmodell berechnen. Diese Berechnung kann mehrere Stunden dauern, wobei während dieser Zeit keine andere VoiceType-Anwendung verwendet werden kann.

6.9.7 Diktierfenster

Das Diktierfenster dient dazu, Diktate zu erfassen. Dabei wird das gesprochene Wort akustisch aufgenommen und in Text umgewandelt. Die Texterfassung kann also nun neben der Tastatur auf über das Mikrophon erfolgen. Das Diktierfenster wurde in der neuen Version von VoiceType um einige Funktionen erweitert und entspricht in der jetzigen Fassung vielmehr einen Editor als einem reinen Fenster zum Erfassen von Text.



Abb. 6.20: Das Diktierfenster

Die Funktion *schnelles Diktat* bietet nicht die Fülle von Funktionen, die das Diktierfenster zur Verfügung stellt, es ist jedoch eine schnelle Möglichkeit, Text in einem Anwendungsprogramm zu erfassen. Diese Funktion wird aktiviert, wenn man in einem Anwendungsprogramm

den Befehl *Diktat starten* gibt. Es öffnet sich ein Fenster, in dem das Diktat erfaßt wird. Nach den Befehl *Diktat stoppen* kann der Text korrigiert und danach direkt in die Zielanwendung übertragen werden.

6.9.8 Editor für Diktiermakros

Ein Diktiermakro ist ein Befehl, der während eines Diktates gesprochen wird und für den ein vordefinierter Text hinterlegt wurde. Wenn der Name des Makros diktiert wird, wird dieser durch den Text ersetzt. Ein Beispiel für ein solches Diktiermakro könnte etwa sein: »Anrede allgemein«. Dieser Text könnte dann durch die folgenden Zeilen ersetzt werden:

»Sehr geehrte Damen und Herren,

«

an dieser Stelle könnte dann der eigentliche Textkörper des Briefes diktiert werden.

Der Diktiermakroeditor kann dazu verwendet werden, derartige Makros oder auch Vorlagen zu erstellen. Vorlagen sind eine besondere Form von Makros, da diese mehrere Bereiche enthalten können, in die Text diktiert werden kann. Mit ihrer Hilfe kann man also Formulare erstellen, die später beim Diktieren ausgefüllt werden.

Der Editor läßt Import und Export von Makros zu. So können Makros und Vorlagen von einem System auf ein anderes übertragen werden. Außerdem können die Makros sortiert oder als Liste ausgedruckt werden.

6.9.9 Benutzerinformationen umstellen

Mit Hilfe dieses Programms können Benutzer von früheren VoiceType Versionen ihre Registrierungen auf die neue Version übertragen. Dabei können das frühere Sprachmodell, die aufgenommenen Wörter und die Diktiermakros, sie mit einem früheren VoiceType erstellt wurden, auch mit der neuen Version verwendet werden.

6.9.10 Zusätzliche Wortschätze installieren / deinstallieren-

Dieses Programm dient dazu, Fachwortschätze für VoiceType zu installieren oder vom System zu entfernen. Sie müssen dazu einen optional erhältlichen Fachwortschatz erwerben, um in einem speziellen Fachgebiet diktieren zu können, ohne daß Sie die für dieses Gebiet üblichen Spezialbegriffe alle aufnehmen und trainieren müssen. Diese Funktion wird vor allem für professionelle Nutzer interessant sein.

Für die deutsche Version sind derzeit folgende Vokabulare für Spezialisten verfügbar:

– Radiologie

– Gynäkologie

– Orthopädie

– Unfallchirurgie

– HNO

– Innere Medizin

– Recht und Wirtschaft

– Technische Gutachten

Für größere Gruppen von Anwendern ist es möglich, einen eigenen Fachwortschatz erstellen zu lassen. Dazu sind etwa 500 Mbyte ASCII-Text des gewünschten Fachgebietes notwendig. So hat sich zum Beispiel die Firma Quelle in Nürnberg einen eigenen Wortschatz für die Kundenbetreuung erstellen lassen und kann so mit höchsten Erkennungsraten arbeiten. Wenn das Erstellen eines Fachwortschatzes für Sie in Frage kommt, wenden Sie sich am besten direkt an IBM.

6.10 Registrierung für Navigation und Diktat

Eine Registrierung (»Enrollment«) zu Beginn der Arbeit mit VoiceType ist prinzipiell nicht notwendig, es gibt jedoch Fälle, in denen eine Registrierung sinnvoll erscheint. Dies gilt, wenn Sie zum Beispiel in einem für VoiceType schwer verständlichen Akzent sprechen oder wenn Sie eine nur geringe Erkennungsleistung des Systems feststellen.

Dennoch möchte ich jedem Benutzer von VoiceType die Registrierung empfehlen!

– Spracherkennungs-»Neulinge«

Sie lernen die Bedienung der neuen Eingabeform. Erinnern Sie sich an Ihre ersten Versuche mit Tastatur oder Maus! Das Training für das Diktieren zwingt Sie zur isolierten Sprechweise. Wenn der Computer etwas falsch versteht, zeigt er es Ihnen an. Wenn Sie sicher sind, daß Sie keinen Fehler gemacht haben, dann lassen Sie sich die Aufnahme wiedergeben: Haben Sie tatsächlich jedes Wort sauber von den anderen getrennt? Wenn es tatsächlich ein Erkennungsfehler von VoiceType war, Sie sprechen aber dieses nicht akzeptierte Wort tatsächlich immer so aus, dann können Sie manuell zum nächsten Satz weitergehen und dann die Registrierung fortsetzen. VoiceType lernt dann diese abweichende Sprechweise trotzdem. Sie sollten dies aber wirklich nur in absoluten Ausnahmefällen tun!!!

– Spracherkennungs-»Profis«

Sie beherrschen die isolierte Sprechweise perfekt, Sie können wie aus der Pistole geschossen diktieren. Allerdings hat sich die verwendete Hardware geändert und vor allem auch die Erkennungssoftware. Die Erkennungsrate ist von Anfang an sehr hoch, wenn Sie das Training gemacht haben, so sind Sie perfekt für das Diktieren vorbereitet. Da Sie schon schnell diktieren können, dauert das Enrollment nur noch eine gute Stunde, diese Zeit können Sie investie-

ren – es lohnt sich.

Sie können allerdings auch Ihre frühere Registrierung von »alten« VoiceType-Versionen migrieren und somit Ihre bereits aufgenommenen Wörter und Diktiermakros weiterhin verwenden.

– Frauen

Das soll nun keine Diskriminierung sein! Es hat sich jedoch herausgestellt, daß VoiceType auf Frauenstimmen nicht immer sehr gut reagiert. Wahrscheinlich waren es vor allem männliche Sprecher, die die akustischen Modelle vorgesprochen haben.

– Tips für alle VTD-Benutzer

Machen Sie nach etwa vier Wochen Arbeit mit VTD ein ganz neues Training, bis dahin haben Sie Routine im Diktieren und sprechen unverkrampft, auch beim Enrollment.

Erstellen Sie Ihr persönliches Sprachprofil in der akustischen Umgebung, in der Sie später die Spracherkennung verwenden werden. Wenn Sie in verschiedenen Umgebungen diktieren werden, so empfiehlt es sich unter Umständen, mehrere Enrollments anzulegen, so z.B. ein Training für zu Hause und eines für ein Büro mit lauten Hintergrundgeräuschen, falls Sie VoiceType auch im Büro verwenden sollten.

6.10.1 Registrierung starten

Nach einem Doppelklick auf das Objekt *Registrierung* wird zunächst ein Fenster angezeigt, in dem Sie Ihren Benutzernamen eingeben, damit Ihre persönlichen Sprachdateien später auch eindeutig identifiziert werden können.

Eine Registrierung ist grundsätzlich ein zweiteiliger Vorgang: zunächst spricht der Benutzer Befehle oder Sätze, die vorgegeben sind, nach. Im zweiten Teil werden aus diesen akustischen Informationen die persönlichen Sprachdateien, die »Registrierung« des Benutzers berechnet. Mit Hilfe dieser Dateien kann VoiceType im Prinzip vorhersagen, wie dieser spezielle Benutzer jedes einzelne Wort im Wortschatz aussprechen wird. Deshalb liegt auch die Erkennungsrate nach einer Registrierung wesentlich höher als vorher, da VoiceType nur auf allgemeine Sprachmerkmale (all der Personen, die den Wortschatz vorgesprochen haben) angewiesen ist, wenn keine Registrierung vorliegt.

Um nun die Registrierung für den gewählten Bereich aufzunehmen, klicken Sie auf die Schaltfläche zum Starten des Trainings. Da im Laufe des Trainings etwa 45 Mbyte an Daten anfallen, testet VoiceType vor Beginn den verfügbaren Festplattenplatz auf der Partition, auf der die Daten abgespeichert werden sollen. Diese Partition können Sie natürlich frei wählen. Wenn nicht genügend Speicherplatz zur Verfügung steht, wird eine entsprechende Warnung ausgegeben und Sie können eine andere Partition wählen. Wenn Sie ein Laufwerk ausgewählt haben, das alle notwendigen Daten aufnehmen kann, wird die Registrierung mit dem ersten Satz gestartet.



Abb. 6.21: Die Benutzerangaben

Im darauffolgenden Fenster für die Registrierung können Sie außerdem wählen, ob Sie die Registrierung für die Navigation oder für das Diktieren durchführen wollen.

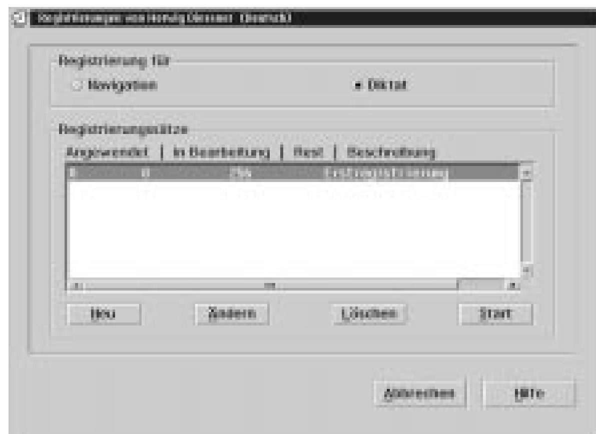


Abb. 6.22: Die Registrierung

6.10.2 Aussetzen und Wiederaufnehmen

Bei der Registrierung handelt es sich um einen längerfristigen Prozeß, und es kann vorkommen, daß Sie eine Registrierung nicht in einem Zug durchführen können. Hier bietet Ihnen VoiceType die Möglichkeit, die Registrierung, das heißt in diesem Falle das Nachsprechen der Wörter, jederzeit zu unterbrechen und zu einem späteren Zeitpunkt wieder aufzunehmen. Der zweite Teil der Registrierung, die Berechnung der persönlichen Sprachdateien kann nicht unterbrochen werden, ohne daß Datenverluste drohen.

6.10.3 Optionen

In den Einstellungen des Registrierungsfensters (Schaltfläche *Optionen*) können Sie festlegen, wie VoiceType bei einer falschen beziehungsweise richtigen Erkennung reagieren soll.



Abb. 6.23: Die Erkennung

So können Sie wählen, ob eine Nachricht angezeigt werden soll, wenn ein Fehler aufgetreten ist oder ob automatisch der nächste Satz angezeigt und die Aufnahme gestartet werden soll, wenn alles korrekt erkannt wurde. Ebenso können Sie die verwendete Soundkarte und deren Anschlüsse auswählen. Mit Hilfe zweier Schieberegler können Sie außerdem den Eingangspegel und die Erkennungsgenauigkeit einstellen.

Tip: Prüfen und ändern Sie gegebenenfalls die Einstellungen, bevor Sie mit dem Sprechen der Sätze anfangen. Wenn Sie die Erkennungsgenauigkeit oder den Eingangspegel mitten während des Aufzeichnens ändern, kann sich dies negativ auf die spätere Erkennungsleistung auswirken.

6.10.4 Kommandos und Sätze aufzeichnen

Wenn Sie die Registrierung für die Navigation durchführen, soll das kontinuierliche Sprechen von Befehlen trainiert werden. Sobald die Aufzeichnung der Akustik gestartet ist, erscheint ein farblich markierter Rahmen um alle Wörter einer Befehlssequenz. Dies bedeutet, Sie müssen alle Wörter kontinuierlich, also ohne Pause zwischen den Wörtern, aussprechen. Alle Wörter, die innerhalb des Rahmens stehen werden also in einem Zug gesprochen.

Anders ist dies bei der Registrierung für das Diktat. Dabei wird immer ein Wort nach dem anderen umrahmt. Sie müssen also nach jedem Wort eine kleine Pause machen. Wenn Sie zum ersten Mal eine Registrierung durchführen, dann ist es günstig, wenn Sie sich genau an den Rahmen halten. Haben Sie später etwas mehr Routine (z. B. wenn Sie nach vier Wochen eine erneute Registrierung durchführen), dann können Sie dem Rahmen auch etwas vorauseilen, immer vorausgesetzt, daß Sie trotzdem noch die Pausen zwischen den Wörtern einhalten.

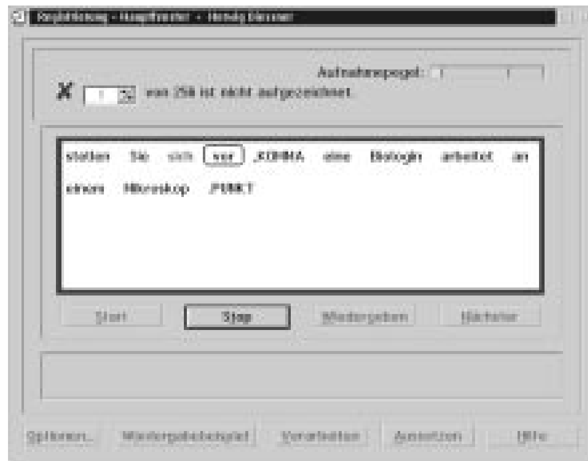


Abb. 6.24: Satz aufnehmen

Um einen Satz aufzunehmen, klicken Sie auf die Schaltfläche *Start*, dann erscheint der Markierungsrahmen und Sie können loslegen. Wenn ein Fehler auftritt können Sie durch *Stop* die Aufnahme unterbrechen, um sie danach zu wiederholen. Mit *Wiedergabe* können Sie sich den Satz genau so wiedergeben lassen, wie Sie ihn gesprochen haben. Damit können Sie bei einer fehlerhaften Aufzeichnung entscheiden, was der Grund für den Fehler war. Mit *Nächster* können Sie zum nächsten Satz weiterblättern, wenn Sie nicht unter *Optionen* eingestellt haben, daß VoiceType automatisch nach der Aufzeichnung zum nächsten Satz weiterblättert.

Bitte achten Sie vor allem bei den ersten Sätzen auf die Aussteuerungsanzeige! Sie darf sich nur im grünen Bereich bewegen. Wenn dies nicht der Fall ist, ändern Sie den Eingangspegel im *Optionen* Fenster.

Vergessen Sie nicht, beim Training für das Diktat jedes – Wort – einzeln – zu – sprechen!

6.10.5 Berechnen der persönlichen Sprachdateien

Nachdem Sie alle Sätze oder Kommandos vorgelesen haben, ist der erste Teil der Registrierung beendet und Sie haben Ihre Aufgabe erledigt. Nun ist der PC an der Reihe und muß aus der riesigen Menge an akustischen Informationen Ihre persönlichen Sprachdateien, Ihre Registrierung berechnen. Je nachdem, wie schnell Ihr PC ist, kann dies mehrere Stunden in Anspruch nehmen. Da dieser Vorgang Ihren Computer sehr in Anspruch nimmt, sollten Sie verzichten, während dieser Zeit mit dem PC nebenher zu arbeiten. Sie könnten zwar auch während der Berechnung den PC nutzen, allerdings wäre dann die Antwortzeit Ihres PC wesentlich schlechter als gewohnt.

Vor dem Start der Berechnung können Sie wählen, ob diese Funktion sofort ausgeführt werden soll oder ob Sie diese zu einem späteren Zeitpunkt, zum Beispiel über Nacht starten wollen.

6.10.6 Registrierungen übertragen

Ein großer Vorteil von VoiceType ist, daß die persönlichen Registrierungen sehr klein sind. Sie passen in der Regel auf eine normale Diskette. So können Sie Ihre Registrierung von dieser einen Diskette auf jeden beliebigen anderen PC mit VoiceType aufspielen und dort weiter diktieren. Dabei hat VoiceType auf dem anderen PC denselben Trainingsstand, wie auf Ihrem eigenen Rechner, mit allen Aussprachen, zusätzlich aufgenommenen Wörtern und Diktiermakros.

Sichern von Registrierungen

Im Hauptfenster für die Registrierung befindet sich eine Schaltfläche, bei deren Anklicken eine Registrierung gesichert wird. Wählen Sie zunächst diejenigen persönlichen Sprachdateien aus, die Sie sichern wollen und klicken Sie auf den Schalter. Sie werden zur Eingabe des Ziellaufwerks und einer Beschreibung für diese Sicherung aufgefordert. Danach werden die Dateien zum Beispiel auf eine Diskette gespeichert.

Wiederherstellen

Zum Wiederherstellen einer Registrierung klicken Sie einfach auf den entsprechenden Knopf und wählen das Laufwerk aus, von dem die Registrierung wiederhergestellt werden soll. Alle dort gefundenen VoiceType Registrierungen werden in einer Liste angezeigt. Aus dieser können Sie die gewünschte Registrierung auswählen und auf das unten auszuwählende Ziellaufwerk installieren lassen. Außerdem können Sie wählen, ob automatisch ein Icon für diese Registrierung auf der Arbeitsoberfläche erstellt werden soll. Wenn der Benutzername dieser Registrierung noch nicht auf dem System vorhanden ist, wird er automatisch angelegt, ansonsten wird die Registrierung überschrieben, wenn Sie die Wiederherstellung nicht abbrechen wollen.

Icon erstellen

Zum Erstellen eines Icons auf der Arbeitsoberfläche öffnen Sie den Ordner *Schablonen* und ziehen das Symbol *Spracheingabebenutzer* auf die Arbeitsoberfläche. Klicken Sie mit der rechten Maustaste auf dieses Icon und wählen Sie aus diesem Kontextmenü die Option *Einstellungen* aus. Hier können Sie nun den Namen des Benutzers oder der Registrierung auswählen und dem Icon einen Namen geben.

6.10.7 Löschen einer Registrierung

Sie können auch auf dem System installierte aber nicht mehr benötigte Registrierungen wieder entfernen, indem Sie einfach den Löschenknopf anklicken, nachdem Sie diejenige Registrierung ausgewählt haben, die Sie löschen möchten. Wenn Sie die Rückfrage bestätigen, werden die persönlichen Sprachdateien des Benutzers gelöscht. Ein eventuell vorhandenes Icon für diese Registrierung wird nicht von der Arbeitsoberfläche gelöscht. Sie müssen dies selbst erledigen.

6.10.8 Zusätzliche Registrierungen

Auf einem PC können sich die Registrierungen mehrerer Personen gleichzeitig befinden. (Es kann aber immer nur ein Benutzer mit VoiceType arbeiten.)

Ebenso kann sich die Registrierung eines Benutzers auf mehreren PC gleichzeitig befinden, wie in Kapitel 6.6 beschrieben.

Darüber hinaus kann ein Benutzer zusätzliche Registrierungen für seine Person erstellen. Das ist dann zum Beispiel sinnvoll, wenn er an zwei unterschiedlichen Orten diktiert, deren akustische Umgebung extrem verschieden ist oder wenn er zwei unterschiedliche Mikrofone benutzt.

Zum Erstellen einer zusätzlichen Registrierung starten Sie einfach das Registrierungsprogramm und wählen Ihre bisher bestehende Registrierung aus. Wenn Sie nun das Training starten können Sie unter Ihrer Person eine weitere Registrierung mit einer anderen Beschreibung erstellen.

6.11 Navigation mit VoiceType

Sie können mit VoiceType die gesamte Oberfläche von OS/2 Warp per Sprache steuern, das heißt, es ist möglich, Programme zu starten, Menüs auszuwählen oder Schaltflächen zu aktivieren oder sogar ganze Befehlssequenzen mit Makros zu automatisieren (vgl. Kapitel über Makros).

6.11.1 Grundsätzliches

Wenn das Mikrofon eingeschaltet ist und sich VoiceType nicht im Diktiermodus befindet, ist automatisch der Modus für Navigation aktiviert. Sie können dann die gesamte Oberfläche per Sprache steuern. Dazu ist es für Sie wichtig, immer zu wissen, auf welches Objekt oder Fenster sich Ihr nächster Befehl beziehen wird. VoiceType hilft Ihnen dabei, dies zu entscheiden: Sobald das Mikrofon eingeschaltet wird, erhält genau ein Fenster einen farblich markierten Rahmen. Die Voreinstellung für den Rahmen ist blau.

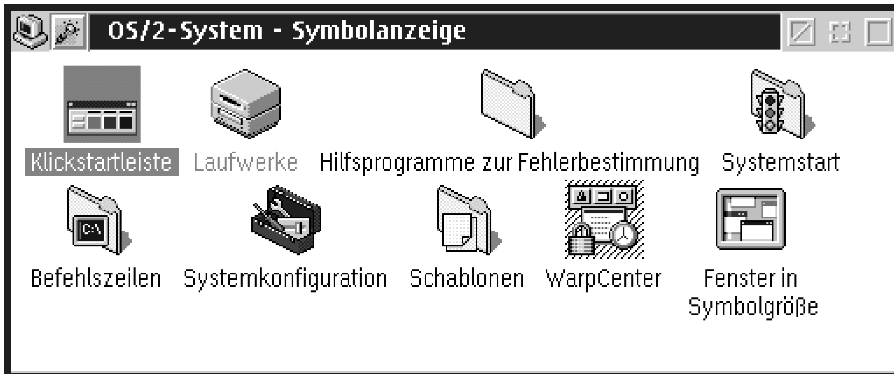


Abb. 6.25: Die verschiedenen Modi

Bevor Sie also mit Ihren Kommandos loslegen, schauen Sie zunächst nach diesem blauen Rahmen. Jeder Befehl, den Sie nun geben, bezieht sich nur auf das markierte Fenster. Um ein anderes Fenster zu aktivieren, können Sie auf das andere Fenster klicken oder mit der Sprache ein paar Befehle geben: »Fensterliste« und danach den Namen des Fensters.

Zusätzlich ändert sich die Titelzeile des aktiven Fensters, denn an der linken Seite wird ein verkleinertes Symbol eingeblendet, das den aktuellen Zustand des Mikrofons anzeigt (ein, aus, Ruhezustand oder Fehler). Im Textteil der Titelzeile wird außerdem der zuletzt erkannte Befehl angezeigt.

Wie bereits erwähnt, erfolgt die Navigation der Oberfläche in kontinuierlicher Sprechweise. Sie müssen also keine Pausen zwischen den Wörtern einhalten, sondern alles wie ein Wort sprechen.

Tip: Beobachten Sie sich selbst, damit Sie nicht in einem monotonen Stimmfall sprechen, wenn Sie alle Wörter aneinanderbinden. Betonen Sie den Befehlssatz so, wie wenn Sie mit anderen Personen sprechen würden. Das hat zwei positive Auswirkungen: erstens, VoiceType erkennt die Befehle besser, zweitens es macht Ihnen, und das ist viel wichtiger, automatisch viel mehr Spaß!

Wenn Sie nach einem möglichen Befehlswort suchen, schauen Sie auf das Fenster *Sprechbare Befehle*. Es legt sich automatisch immer über alle anderen Fenster und somit haben Sie ständig eine kleine Gedächtnisstütze für Ihre Kommandos. Außerdem werden in diesem Fenster auch ungewöhnliche Befehle, die sich nur auf ein spezielles Programm beziehen angezeigt.

6.11.2 Sprechbare Befehle

Wie bereits erwähnt, zeigt dieses Fenster die aktuell möglichen Befehle zur Oberflächensteuerung an. Diese Befehle gelten dann nur für dasjenige Fenster, das durch den blauen Rahmen markiert ist.

Die Befehle sind in zwei Gruppen aufgeteilt: Wörter, die sich speziell auf ein bestimmtes Programm beziehen und generische Wörter, das sind Befehle, die generell für das gesamte Betriebssystem Gültigkeit haben.

Innerhalb dieser Gruppen können Sie selbst wählen, welche Arten von Wörtern Ihnen im *Sprechbare Befehle*-Fenster angeboten werden sollen. Dazu öffnen Sie das Menü *Aktionen* und wählen entweder *Generische Wörter* oder *Programmspezifische Wörter* aus. Ein Fenster wird dargestellt, in dem Sie die Kategorien der darzustellenden Befehle selektieren können.



Abb.6.26: Befehlskategorien

Tip: Wählen Sie nicht zu viele Kategorien aus, damit Sie nicht den Inhalt des Fensters mit seiner vertikalen Schiebeleiste verschieben müssen. Beim Verschieben ändert sich der blaue Fokus und damit auch gleichzeitig der Inhalt des *Sprechbare Befehle*-Fensters. Damit können Sie nie die verdeckten möglichen Befehle für ein bestimmtes Programm sehen.

Es kann vorkommen, daß ein Wort im Fenster durchgestrichen angezeigt wird. Dies bedeutet, daß VoiceType diesen Befehl als einen solchen erkannt hat, jedoch keine entsprechende Aussprache zuordnen kann. Sie müssen also eine Aussprache für diesen Befehl trainieren, wenn Sie den Befehl verwenden wollen. Dazu wählen Sie den durchgestrichenen Befehl aus und klicken auf die Schaltfläche (oder sprechen) *Trainieren*. In dem daraufhin dargestellten Fenster können Sie die Akustik aufnehmen. Tips zu diesem Vorgang können Sie im Kapitel *Befehle trainieren* nachlesen.

6.11.3 Programme starten

Wenn sich ein Programmobjekt auf der Arbeitsoberfläche befindet, können Sie es starten, indem Sie folgende Befehle sprechen (sofern der Name nicht im *Sprechbare Befehle* durchgestrichen angezeigt wird):

»Arbeitsoberfläche«

»Programm«

»öffnen«, wobei Sie für *Programm* den Namen des Programms einsetzen.

Eine andere Möglichkeit bietet sich, wenn Sie das *Sprechbare Titel*-Fenster nutzen. Sagen Sie: »Sprechbare Titel«. Das Fenster wird angezeigt. Sein Inhalt besteht aus den Symbolen aller Programme, die Sie per Sprache starten können. »Sprung zu« lautet Ihr nächster Befehl, und *Programm* wird gestartet.

Um ein bereits aktives aber nicht sichtbares Programm darzustellen sprechen Sie »Fensterliste«, um zunächst die Fensterliste anzuzeigen. Nun können Sie durch *zu Programmwechseln* das Programm auf den Bildschirm bringen. Sie können die Auswahl in der Fensterliste ändern, indem Sie Richtungsbefehle geben. »Auf«, »Ab« oder auch »Auf« – »zwei«, »Ab« – »fünf«.

6.11.4 Menüs aufrufen

Hier zeigt VoiceType, zu welchen Leistungen dieses System in der Lage ist. Egal welches Programm Sie nutzen, sprechen Sie einfach ein Wort, das Sie in der Menüleiste finden, und VoiceType öffnet das Auswahlmenü. Wenn Sie aus dieser Auswahl wieder einen Befehl sprechen, wird dieser sofort ausgeführt. Haben Sie etwas Routine und wissen, welche Befehle in diesen Untermenüs zur Verfügung stehen, können Sie diese auch direkt abgeben und VoiceType kann sie ausführen.

Nun aber der eigentliche Clou: VoiceType scannt also die Menüstruktur des aktiven Anwendungsprogrammes. Stellen Sie sich nun vor, diese Menüstruktur wurde dynamisch programmiert, daß sie sich während der Programmlaufzeit ständig ändern kann. Auf all diese Menüpunkte haben Sie als Befehl Zugriff.

Genau das nützt der in OS/2 Warp 4 integrierte Browser für das World Wide Web aus: Sie haben ein WWW Dokument mit sogenannten Hotlinks geladen. Der Browser ist so programmiert, daß er dieses Dokument nach den Hotlinks durchsucht und diese dynamisch in die Menüzeile aufnimmt. Sie können nun beim Lesen des Dokuments einfach einen Hotlink aussprechen, VoiceType ist in der Lage, diesen Link als Befehl zu erkennen und ruft automatisch das hinter dem Link liegende Dokument auf. So ist es für Sie möglich, durch das Web zu surfen, ohne auch nur einen Finger krumm zu machen, denn es geht alles mit Ihrer Stimme.

6.11.5 Cursor bewegen

Sie können natürlich auch innerhalb eines Textes navigieren und den Cursor an eine bestimmte Stelle im Text positionieren oder Wörter, Zeilen und ganze Textblöcke markieren. Um diese

»Grammatik« für ein bestimmtes Programm zu aktivieren, öffnen Sie die Einstellungen des Programms, wechseln Sie zur dritten Seite der *Spracheingabe* und aktivieren Sie die Option *Texteditierung*. Die Befehle, die Sie zum Bewegen im Text verwenden können, stehen Ihnen sowohl in dem von Ihnen gewählten Programm (zum Beispiel dem OS/2 Systemeditor) als auch auf der Arbeitsoberfläche zur Verfügung. So können Sie schnell Symbole der Arbeitsoberfläche anspringen oder auswählen.

6.11.6 Fenster manipulieren

Fenster können mit Spracheingabe verschoben, vergrößert oder verkleinert werden. »Maximale Größe« maximiert ein Fenster. Dies können Sie mit »Wiederherstellen« rückgängig machen. »Unterer Rahmen« – »abwärts« vergrößert ein Fenster, indem es nach unten verlängert wird. Die weiteren Befehle, auch wie Sie mit kontinuierlicher Sprache Fenster manipulieren können, finden Sie in der Onlinehilfe.

6.11.7 Navigation in Texten

Um Texte zu scrollen, die nicht mehr auf eine Bildschirmseite passen, werden im Fenster horizontale und vertikale Schiebeleisten eingeblendet. Bisher haben Sie diese immer mit der Maus verschoben, nun können Sie es auch mit Befehlen: »Vertikale Schiebeleiste« – »abwärts« – »Stop«, so sieht eine beispielhafte Kommandofolge aus.

6.11.8 Buchstabiermodus und Zahlenmodus

Buchstabieren können Sie nicht nur, wenn Sie Texte erfassen. Auch bei der Steuerung kann Ihnen der Buchstabiermodus nützliche Hilfe leisten. So können Sie nun in jedes Eingabefeld direkt buchstabieren. Selbst in einem OS/2 oder DOS Fenster können Sie durch Buchstabieren Kommandozeilen eingeben. Daß dies natürlich nicht immer die beste Methode ist, Felder auszufüllen, liegt auf der Hand, aber man kann so nahezu vollständig auf die Tastatur verzichten.

In einen OS/2- oder DOS-Gesamtbildschirm kann man nicht buchstabieren, die Sprachsteuerung ist auf die Arbeitsoberfläche begrenzt.

»Buchstabieren starten« schaltet in den Buchstabiermodus, »Zurück« beendet ihn wieder. Sie können mit Buchstaben Groß-A – u – t – o, mit dem phonetischen Alphabet Groß-Anton – Ulrich – Theodor – Otto oder mit einem Gemisch aus beidem buchstabieren. Wird ein Buchstabe zum Beispiel »E« falsch erkannt, können Sie ihn korrigieren, indem Sie sagen »wie Emil«. Der falsche Buchstabe wird dann gelöscht und durch einen neuen ersetzt.

Ebenso können Sie den »Zahlenmodus« mit »Zahlen starten« aktivieren, um Ziffern zu erfassen. Das Zahlenvokabular umfaßt dabei die Ziffern 0 bis 9 und alle Satzzeichen. Die zuletzt diktierte Zahl können Sie durch »Rückgängig« löschen, »Leerzeichen« fügt ein Leerzeichen zwischen zwei Ziffern ein.

Buchstabieralphabet

VoiceType erkennt das phonetische Alphabet. Es besteht aus folgenden Wörtern:

a: Anton

b: Berta

c: Caesar

ch: Charlotte

d: Dora

e: Emil

f: Friedrich

g: Gustav

h: Heinrich

i: Ida

j: Julius

k: Konrad

l: Ludwig

m: Martha

n: Nordpol

o: Otto

p: Paula

q: Quelle

r: Richard

s: Siegfried

t: Theodor

u: Ulrich

v: Viktor

w: Wilhelm

x: Xaver

y: Ypsilon

z: Zacharias

ä: Ägypten

ö: Ökonom

ü: Übermut

ß: Scharfes-S

6.11.9 Befehle trainieren

Ein Wort, das im Fenster *Sprechbare Befehle* durchgestrichen angezeigt wird, kann nachträglich dem Befehlswortschatz hinzugefügt werden, damit Sie diesen Befehl in Zukunft per Sprache ausführen können. Klicken Sie dazu zweimal auf das entsprechende Wort, um einen »Rekorder« für die Akustik zu öffnen.

Wenn der Befehl nur aus einem Wort besteht und dieses so ausgesprochen wird, wie es geschrieben wurde, können Sie sofort auf dem Aufnahmeknopf klicken und die Aussprache aufnehmen.

Besteht der Befehl jedoch aus mehreren Wörtern oder muß man das Wort anders aussprechen als seine Schreibweise (so zum Beispiel fremdsprachliche Wörter), so sollten Sie zunächst auf die Schaltfläche *Spezial* klicken, um den erweiterten Rekorder zu öffnen.



6.27: Der erweiterte Rekorder

Bei mehreren Wörtern klicken Sie auf *Ganze Ausdrücke* und nehmen Sie den gesamten Befehl als ein Wort auf. Da dadurch mehr akustische Information als eine Einheit für VoiceType zur Verfügung steht, wird dieser Befehl von der Spracherkennung leichter erkannt.

Wollen Sie beispielsweise den Befehl *Close* für ein englischsprachiges Programm aufnehmen, so geben Sie in der Klangbeschreibung eine möglichst genaue Nachbildung der Aussprache in deutschen Buchstaben ein, etwa »Klous«, und nehmen Sie dann das Wort akustisch auf.

Beachten Sie, daß Wörter in Großbuchstaben im Rekorder mit Bindestrichen dargestellt werden. VoiceType erwartet nun, daß der Befehl buchstabiert wird. Wollen Sie dennoch die »normale« Aussprache aufnehmen, so löschen Sie einfach im Feld für die akustische Beschreibung alle Bindestriche aus dem Wort.

Wenn VoiceType die Aussprache nicht erkannt hat, so wird ein Warnhinweis ausgegeben, der zu einer erneuten Aufnahme auffordert. Sobald VoiceType die Aussprache erkannt hat, ist der Befehl im *Sprechbare Befehle* Fenster nicht mehr durchgestrichen und Sie können das Kommando sofort anwenden.

Wenn Sie mehrere Aussprachen hintereinander aufnehmen wollen, so öffnen Sie die Einstellungen der VoiceType Sprachsteuerung und wählen dort die Seite Aussprache aus. Auf dieser Seite werden alle Wörter angezeigt, die VoiceType nicht in ihrer Aussprache kennt. Sie können hier also diejenigen Wörter auswählen, die Sie trainieren möchten.

Sie können auch für Wörter, die bereits von VoiceType erkannt werden eine neue Aussprache aufnehmen. Dabei haben Sie die Wahl, ob diese neue Aussprache die alte ersetzen oder alternativ zu der alten verfügbar sein soll. So können Sie etwa statt »Ruhezustand einleiten« »Geh schlafen« trainieren oder statt »Ruhezustand Ende« »Wach auf« sagen.

6.11.10 Ruhezustand

Das Mikrofon kann durch Mausklick oder zweimaliges Drücken der Umschalt-Taste (Shift) ein- oder ausgeschaltet werden. Möchten Sie auch selbst auf diese »manuellen« Eingaben verzichten, so nutzen Sie am besten den Ruhezustand. Sagen Sie einfach bei eingeschaltetem Mikrofon »Ruhezustand einleiten« und das Mikrofon wird sozusagen »schlafen gelegt«. Es reagiert nun nur noch auf zwei Befehle: »Ruhezustand Ende« oder »Mikrofon aus«.

Wenn sich VoiceType im Ruhezustand befindet, wird der Ressourcen-Bedarf der Spracherkennung gesenkt, liegt jedoch höher als bei ausgeschaltetem Mikrofon.

Sie können die Einstellungen von VoiceType so verändern, daß beim Start der Spracherkennung das Mikrofon automatisch in den Ruhezustand geht. So können Sie jederzeit durch einen gesprochenen Befehl die Sprachsteuerung und das Diktat aktivieren.

6.12 Diktieren mit VoiceType

Die Hauptfunktion, die Ihnen VoiceType bietet, ist das Erfassen von Text durch Ihre gesprochenen Wörter. Aus dieser Funktion hat VoiceType auch seinen Namen erhalten: »Tippen mit der Stimme« könne man es vereinfacht nennen. Böswillig ausgedrückt könnte man sagen, die Navigation der Oberfläche von OS/2 Warp 4 ist nur ein »Abfallprodukt« der jahrelangen Forschung und Entwicklung zur Umsetzung von Sprache in Schrift. In dieser Disziplin ist VoiceType durch seine einzigartige Erkennungstechnik extrem leistungsfähig und beinahe ohne Konkurrenz.

Es gibt zwei Möglichkeiten, durch Ihre Sprache Texte zu erfassen:

– Diktierfenster

Das Diktierfenster bietet Ihnen sämtliche Funktionen eines Editors zur Manipulation von Texten. Sie können Texte durch Diktat erfassen, formatieren und nachbearbeiten und dann als ASCII-Datei abspeichern oder direkt in andere Anwendungen übergeben. Mit dem Befehl »Diktierfenster« können Sie diesen Editor jederzeit aufrufen.

– Schnelles Diktat

Möchten Sie »mal eben« einen Text in eine Anwendung diktieren, so sagen Sie einfach »Diktat starten«, und das Fenster für das schnelle Diktat wird geöffnet. Dieses bietet Ihnen nicht den großen Funktionsumfang des Diktierfensters, aber Sie können hierin den Text erfassen, korrigieren und danach den Text direkt in Ihre Zielanwendung einbinden.

Welche Art des Diktates auch immer Sie wählen, es ist immer wichtig, daß Sie den diktierten Text korrigieren, bevor Sie den Text übertragen oder abspeichern. VoiceType lernt aus den Korrekturen und wird so immer besser in der Erkennung. Andererseits lernt es falsch, wenn Sie Fehler nicht korrigieren und wird dann immer schlechter. Das heißt also für Sie, daß Sie nach dem Diktat zuerst gewissenhaft den Text noch einmal durchgehen und alle fehlerhaften Erkennungen korrigieren müssen.

6.12.1 Grundsätzliches zum Diktieren

Bevor Sie sich in »Medias Res« stürzen, hier noch ein paar Tips. Wenn Sie diese Punkte beachten, werden Sie schnell und einfach mit dem Diktieren vertraut sein.

- Nachdem Sie »Diktat starten« ausgesprochen haben, warten Sie einen Augenblick, bis im Fenster »Bereit zum Diktat...« erscheint. Legen Sie erst dann mit Ihrem Diktat los.
- Das *Sprechbare Befehle* Fenster zeigt in diesem Zustand sämtliche Diktiermakros an, die Sie während des Diktats verwenden können.
- Denken Sie daran, nach jedem Wort eine kleine Pause zu machen, etwa eine Zehntelsekunde genügt.
- Sprechen Sie die einzelnen Wörter so, wie Sie es gewöhnt sind. Versuchen Sie nicht etwa, besonders deutlich (und damit für Sie unnatürlich) zu sprechen.
- Satzzeichen werden mitdiktiert: »Punkt«, »Komma«, »Ausrufezeichen«, »Fragezeichen« und so weiter.
- Formatierungen können mitdiktiert werden: »Neuer Absatz«, »Neue Zeile«, »Bindewort«, wenn zwei Wörter zusammengeschrieben werden sollen. »Großschreibung« muß nach Punkt, Fragezeichen, Ausrufezeichen oder nach einem Absatz nicht mitdiktiert werden.
- Zerlegen Sie beim Diktat große Zahlen in Zweierblöcke: 568412 wird so diktiert: »56« – »84« – »12«.
- Befehle für Diktiermakros werden als ein Wort ausgesprochen »Brief-beenden«.
- VoiceType prüft bereits während des Diktats den Kontext. So kann es vorkommen, daß zunächst ein völlig unpassendes Wort geschrieben wird. Lassen Sie sich dadurch nicht verwirren, und diktieren Sie weiter. Mit den weiter diktierten Wörtern kann VoiceType erneut den Kontext prüfen und dann von selbst das zunächst falsche Wort wieder verändern. Am besten schauen Sie während des Diktats gar nicht auf den Bildschirm, sondern konzentrieren sich auf Ihre Vorlagen, Aufzeichnungen oder auf das Formulieren der Sätze.

- Nach dem »Diktat stoppen« können Sie sich dem Computer wieder zuwenden, denn nun muß die Korrektur erfolgen. Sie müssen sich das Diktat und die Korrektur als einen einzigen Vorgang vorstellen. Kein Diktat ohne Korrektur! Nur so wird VoiceType von Mal zu Mal besser.
- Sämtliche Korrekturen werden beim Speichern oder Übertragen des Textes in Ihr Sprachmodell aufgenommen. Neu aufgenommene Wörter und ihre Aussprache werden in Ihre persönlichen Sprachdateien eingefügt.
- Durch eine Registrierung können Sie von Anfang an mit noch besseren Erkennungergebnissen von VoiceType rechnen. Führen Sie also ein Training durch und profitieren Sie von guten Erkennungsergebnissen!
- Da Ihre persönlichen Sprachdateien ständig aktualisiert werden, ist es sinnvoll, regelmäßig Sicherungskopien dieser Dateien anzulegen. Führen Sie dazu das Sichern und Wiederherstellen von Registrierungen wie im Kapitel über Registrierungen beschrieben durch.

6.12.2 Wörter korrigieren

Nach dem Diktat muß die Korrektur erfolgen. Dazu können Sie sich Teile des Textes akustisch wiedergeben lassen und die erkannten Wörter überprüfen. Markieren Sie dazu den Text mit der rechten (!) Maustaste. Automatisch wird ein Menü angezeigt, in dem Sie das Diktat akustisch über den Lautsprecher wiedergeben lassen können. Beachten Sie dabei, daß Sie maximal etwa 100 Wörter wiedergeben lassen können. Wenn Sie mehr Text auswählen, können Sie die Wiedergabeoption nicht anwählen.



Abb. 6.28: Wiedergabeoption

Markieren Sie den Text und hören Sie sich die Akustik noch einmal an. Wenn Sie einen Fehler finden, brechen Sie die Wiedergabe ab. Klicken Sie mit der rechten Maustaste auf das Wort, das falsch erkannt wurde. Haben Sie sich versprochen? Haben Hintergrundgeräusche das Signal verfälscht? Hat VoiceType das Wort falsch verstanden? Je nachdem, weshalb der Fehler aufgetreten ist, müssen Sie unterschiedlich korrigieren:

– Sie haben falsch gesprochen oder Hintergrundgeräusche haben die Akustik verfälscht.

In diesem Fall markieren Sie das Wort mit der rechten (!) Maustaste und klicken auf *Löschen*. Sagen Sie »Diktat starten«, und diktieren Sie das Wort erneut. Auch wenn unter Umständen das manuelle Eintippen schneller gehen könnte, so hat das erneute Diktieren einen enormen Vorteil: Da VoiceType Wörter in ihrem Kontext prüft und statistisch erfaßt, lernt das System die Verwendung dieses Wortes in seinem Kontext. So kann es beim nächsten Diktieren um so besser erkennen.

– VoiceType hat einen Fehler gemacht

Es liegt nun an Ihnen zu entscheiden, welcher Art dieser Fehler ist. Es könnte sein, daß ein einzelnes Wort, das Sie richtig gesprochen haben falsch aufgeschrieben wurde. Sie klicken mit der rechten Maustaste auf das Wort, oder sagen Sie »Fehler korrigieren«. Es wird automatisch markiert und die Akustik wiedergegeben. Gleichzeitig wird das Korrekturfenster geöffnet, in dem Ihnen Alternativvorschläge zur Auswahl angeboten werden. Befindet sich das richtige Wort in dieser Liste, können Sie es per Mausklick auswählen oder durch Sprechen der vorangestellten Zahl. Befindet sich das richtige Wort nicht in der Liste, so klicken Sie auf *Fehler korrigieren* und tippen Sie das Wort einfach ein oder starten Sie den Buchstabiermodus, um das Wort zu buchstabieren. Sie können das neue Wort auch sofort eingeben, nachdem Sie das falsche mit der rechten Maustaste markiert haben. Wenn das richtige Wort nicht im Wortschatz enthalten ist, versucht VoiceType, es automatisch aufzunehmen. Sie erhalten eine Nachricht, ob das Wort zum Wortschatz hinzugefügt wurde oder nicht. Falls nicht, können Sie die Korrekturweise *Spezial* verwenden, die automatisch ein Korrekturfenster anzeigt, wenn das Aufnehmen des Wortes fehlgeschlagen ist. Sie müssen dann zusätzlich zu der Schreibweise des Wortes auch eine akustische Beschreibung eingeben. Dies wird vor allem bei fremdsprachlichen Lehnworten vorkommen, zum Beispiel beim Wort »VoiceType«, dessen akustische Beschreibung in etwa so aussehen könnte: »Woistaip«. Hier wird also versucht, mit den im deutschen vorkommenden Lauten und mit deutschen Buchstaben den Klang des Wortes nachzubilden.

Wenn Sie ein Wort gesprochen haben, VoiceType aber zwei Wörter aufgeschrieben hat, so markieren Sie beide Wörter mit der rechten Maustaste und tippen oder buchstabieren das richtige Wort.

Haben Sie hingegen zwei Wörter gesprochen und VoiceType hat nur ein Wort aufgeschrieben, so war die Pause zwischen den beiden Wörtern zu kurz. Markieren Sie das falsche Wort mit der rechten Maustaste, löschen Sie e,s und diktieren Sie die beiden Wörter noch einmal.

Es könnte auch sein, daß das Wort an sich zwar richtig erkannt, aber falsch formatiert wurde. Das Korrekturfenster bietet Ihnen die Option *Format* an. Damit können Sie Groß- und Kleinschreibung wechseln und Zahlen als Ziffern oder Buchstaben darstellen lassen. Ebenso können Sie auswählen, daß ein Wort komplett in Großbuchstaben geschrieben wird.

6.12.3 Text markieren und löschen

Wenn Sie Manipulationen am diktierten Text vornehmen wollen, müssen Sie die entsprechenden Bereiche zunächst markieren. Dies könnten Sie wie bisher mit der Maus durchführen. Doch jetzt haben Sie ja auch die Möglichkeit mit Befehlen zu arbeiten:

»Wort-auswählen«

»Zeile-auswählen«

»Alles-auswählen«

»Auswahl-zurücknehmen«

»Zeile-löschen«

»Alles löschen«

Dies sind die Befehle, die Ihnen neu zur Verfügung stehen. »Alles-auswählen« kann unter Umständen etwas gefährlich sein zu verwenden, da zwar selten, aber doch einmal der Befehl von VoiceType falsch »verstanden« und mit »Alles-löschen« verwechselt werden könnte.

6.12.4 Text editieren

Wie in einem »normalen« Editor können Sie Text auch von Hand eintippen. Dieser Text wird jedoch nicht in das Sprachmodell von VoiceType aufgenommen und wirkt sich auch nicht auf die Kontextprüfung aus. Sie sollten deshalb falsch erkannte Wörter nicht einfach löschen und von Hand eintippen sondern wirklich mit »Fehler korrigieren« verbessern.

Verwenden Sie andererseits nicht »Fehler korrigieren« um Änderungen am Text vorzunehmen, denn sonst lernt VoiceType etwa für das Wort »und« die Aussprache von »oder«, nur weil Sie diese Änderung durchführen wollten, ohne neu zu diktieren. Sie können sich vorstellen, was für Texte entstehen, wenn für einige Wörter völlig andere Aussprachen gespeichert sind.

6.12.5 Arbeit mit dem Diktierfenster

Das Diktierfenster erscheint auf den ersten Blick wie ein ganz gewöhnlicher Editor. Bei näherem Hinsehen erkennt man jedoch, daß diese Anwendung auf Funktionen der Spracherkennung optimiert ist und bisher nicht dagewesenes zu leisten vermag. Die wichtigesten Funktionen des Diktierfensters sind nicht nur über das Menü zugänglich sondern können direkt über Symbole angewählt werden.



6.29:
Das Diktierfenster

Text übertragen

Wenn Sie den Befehl »Text übertragen« geben, wird der markierte Text aus dem Diktierfenster ausgeschnitten und in Ihre Zielanwendung kopiert. Wurde kein Bereich im Text markiert, wird standardmäßig der gesamte Text übertragen. Sie können diese Voreinstellungen ändern, indem Sie unter *Optionen* die *Übertragungsoptionen* anpassen.

Nach dem Befehl *Text übertragen* ändert sich der Mauszeiger in eine Sprechblase. Klicken Sie nun mit der Maus auf Ihre Zielanwendung, um den Text einzufügen. Möchten Sie den Transfer abbrechen, drücken Sie einfach die Esc-Taste. Denken Sie bei der Übertragung daran, daß VoiceType nur richtig lernt, wenn der Text im Diktierfenster korrigiert wurde. Eine Korrektur in der Zielanwendung aktualisiert nicht Ihre Sprachdateien!

Text abspeichern

Wenn Sie über das Menü *Datei|Speichern* oder *Speichern unter* den diktierten Text abspeichern wollen, denken Sie bitte daran, daß zu diesem Zeitpunkt Ihre persönlichen Sprachdateien aktualisiert werden. Deshalb sollten Sie nur vollständig korrigierte Diktate abspeichern. Wenn Ihnen eine Korrektur aus Zeitmangel nicht sofort möglich ist, lesen Sie doch den nächsten Punkt.

Diktiersitzungen für spätere Korrekturen abspeichern

Eine bisher noch nicht in VoiceType verfügbare Funktion ist das Abspeichern einer Diktiersitzung. Wenn Sie das Upgrade für VoiceType oder einen speziellen Fachwortschatz erwerben, können Sie diese Funktion nutzen.

Dabei wird der diktierte Text zusammen mit der aufgenommenen Akustik abgespeichert. Das persönliche Sprachmodell wird jedoch nicht verändert. Sie können so die Korrektur des Textes später durchführen, ohne in die Gefahr zu kommen, daß VoiceType durch das Abspeichern

eines unkorrigierten Textes fehlerhaft lernt. Zusammen mit dieser Diktiersitzung können Sie auch eine etwa 10 Sekunden lange Nachricht abspeichern. Dies ist zum Beispiel dann sinnvoll, wenn Sie nicht selbst den Text korrigieren und nachbearbeiten werden.

Nachdem Sie das Diktat beendet haben, klicken Sie auf *Diktat|Sitzung sichern*. Sie können in diesem Fenster den Namen und den Pfad der Datei, aber auch eine Beschreibung der Diktiersitzung eingeben. Ebenso können Sie eine akustische Nachricht aufnehmen, die zusammen mit der Diktiersitzung abgespeichert wird.

Diese Diktiersitzung kann zu einem späteren Zeitpunkt über *Diktat|Sitzung öffnen* wieder in das Diktierfenster zur Korrektur eingelesen werden. Mit dem Menüpunkt *Diktat|Akustische Beschreibung* kann nun die zusätzliche Nachricht abgehört werden.

6.12.6 Optionen für das Diktierfenster

Sie können das Diktierfenster und seine Verhaltensweise bei Diktaten weitgehend nach Ihren persönlichen Präferenzen konfigurieren. Dazu dient vor allem der Menüpunkt *Optionen*.

Sie können auswählen, ob die Symbolleiste dargestellt werden soll oder nicht, doch die größeren Gestaltungsmöglichkeiten haben Sie mit den Optionen für Diktat und Textübertragung.

Bei den Optionen für das Diktieren können Sie folgende Punkte konfigurieren:

– Diktieren im Hintergrund: Diese Option erlaubt Ihnen, ein Diktat weiterzuführen, auch wenn Sie inzwischen ein anderes Fenster selektiert oder geöffnet haben. Standardmäßig ist diese Option ausgeschaltet. In der Praxis wird es auch nur sehr wenige Anwendungsfälle geben, in denen das Aktivieren dieser Option wirklich sinnvoll ist.

– Aufgenommene Wörter rechtschreibprüfen: Bei eingeschalteter Option werden alle korrigierten Wörter auf ihre Rechtschreibung hin überprüft, bevor sie in den Wortschatz von VoiceType übernommen werden.

– Anzahl der Alternativvorschläge: Wenn Sie ein falsch erkanntes Wort korrigieren, wird Ihnen eine Liste von Alternativen angeboten. Sie können an dieser Stelle einstellen, wie viele Alternativen angezeigt werden sollen, Sie haben die Wahl zwischen 0 und 16. Standardeinstellung ist 8 Auswahlmöglichkeiten.

– Zahlenmodus erlauben: Wenn Sie diese Option auswählen, können Sie während des Diktates den Zahlenmodus aktivieren, um die Darstellung von Zahlen als ausgeschriebene Wörter zu unterbinden.

– Buchstabiermodus erlauben: Analog können Sie hiermit den Befehl *Buchstabieren starten* aktivieren und so zum Beispiel Eigennamen, die Sie nicht in den Wortschatz aufnehmen wollen, durch Buchstabieren erfassen.

Die Optionen für die Textübertragung in die Zielanwendung können ebenfalls genau definiert werden.

– Automatische Auswahl: Wenn Sie vor dem Übertragen des Textes keinen Bereich ausgewählt haben, wird der gesamte Text automatisch ausgewählt.

– Zeilenumbrüche erhalten: Der übertragene Text erhält bei aktivierter Option in der Zielanwendung genau dieselben Zeilenumbrüche wie im Diktierfenster. Standardmäßig ist diese Option inaktiv.

– Übertragungsart: An dieser Stelle können Sie bestimmen, wie die Übertragung des Textes durchgeführt werden soll. Wenn Sie Probleme beim Übertragen in bestimmte Anwendungen haben, so ändern Sie hier die Einstellungen, um eventuell eine Lösung zu finden. Die Übertragung findet in zwei Schritten statt.

Zunächst wird der Text vom Diktierfenster in die Zwischenablage übertragen.

Mit *Ausschneiden* wird der Text beim Übertragen aus dem Diktierfenster entfernt und in die Zwischenablage gespeichert. Das Diktierfenster ist danach leer und Sie können weitere Diktate starten.

Kopieren überträgt den Text in die Zwischenablage, wobei das Diktat zusammen mit Text und Akustik im Diktierfenster verbleibt.

Im zweiten Schritt wird der Text aus der Zwischenablage in die Zielanwendung eingefügt.

Einfügen ist die Standardauswahl. Der Text wird aus der Zwischenablage als ein Block in die Zielanwendung eingefügt. Vorteil dieser Methode ist, daß das Übertragen sehr schnell vonstatten geht. Ein Nachteil kann sein, daß manche Anwendungen den ankommenden Text nicht so schnell verarbeiten können.

Zeichenweise den Text zu übertragen bedeutet, daß der Text sozusagen als viele einzelne Tastenanschläge übertragen wird. Diese Übertragung benötigt wesentlich mehr Zeit, hat aber den Vorteil, daß sie mit beinahe allen Anwendungen funktioniert. Ein guter Anwendungsfall wäre die Übertragung von Diktaten in eine 3270-Emulation, da die Host-basierten Systeme in der Regel nur einzelne Tastenanschläge verarbeiten können.

6.12.7 Schnelles Diktat

Sie können aus jeder Anwendung heraus das »Schnelle Diktat« starten, um Text direkt für diese Applikation zu erfassen. Sagen Sie einfach »Diktat starten« und das Fenster für das »Schnelle Diktat« wird geöffnet. Auch dieses Fenster bietet weitgehend die Konfigurationsmöglichkeiten wie das Diktierfenster. Unter *Optionen* können Sie dieselben Optionen auswählen wie im vorherigen Teil beschrieben. *Makros* ruft den Diktiermakro-Editor auf.

Wenn Sie mit Ihrem Diktat fertig sind und alle Fehler korrigiert haben, klicken Sie auf die Schaltfläche *Senden*, um den Text in die Zielanwendung zu übertragen. Das Fenster wird dabei nicht geschlossen, damit es für weitere schnelle Diktate aus dieser Anwendung zur Verfügung steht. Sie können jedoch das Fenster duch einen Mausklick schließen, wenn Sie kein weiteres Diktat erfassen möchten.

Tip: Aus Sicherheitsgründen legt VoiceType grundsätzlich von den letzten drei Diktaten im Fenster *Schnelles Diktat* Kopien an. Diese befinden sich im Verzeichnis \VT\SPCH_RUN\ und tragen den Namen Dictatex.Doc*, wobei das x für eine Ziffer steht. Falls Ihnen tatsächlich einmal ein Diktatstext verlorengehen sollte, können Sie dieses unter Zuhilfenahme der genannten Dateien rekonstruieren.*

6.13 VoiceType-Einstellungen anpassen

Je nachdem, wie vertraut Sie mit dem Spracherkennungssystem bereits sind, können Sie verschiedene Einstellungen ändern, um VoiceType ideal an Ihre Wünsche und Bedürfnisse anzupassen. In diesem Kapitel sollen nun sämtliche Optionen beschrieben werden. Um dies zu tun, öffnen Sie das Fenster *Eigenschaften* der VoiceType-Sprachsteuerung.

6.13.1 VoiceType automatisch starten

Wenn Sie möchten, daß VoiceType automatisch bei jedem Systemstart aufgerufen wird, erstellen Sie eine Referenz des Objektes Ihrer Registrierung und legen Sie diese Referenz in den *Systemstart*-Ordner. Bei dem nächsten Start von OS/2 Warp 4 wird dann VoiceType automatisch gestartet.

Außerdem können Sie VoiceType so konfigurieren, daß beim Start der Ruhezustand aktiviert wird. Nach dem gesprochenen Befehl »Ruhezustand Ende« haben Sie dann Zugriff auf sämtliche Funktionen der Spracherkennung, ohne daß Sie auch nur eine Taste drücken mußten.

6.13.2 Einstellungen der Sprachsteuerung

Wenn Sie das *Eigenschaften*-Fenster der Sprachsteuerung geöffnet haben, sehen Sie zunächst die erste von drei Seiten *Optionen*. Die ersten beiden Seiten beziehen sich auf die Sprachsteuerung selbst, die dritte Seite dient zur Konfiguration des *Sprechbare Befehle*-Fensters.

Auf der ersten Seite können sieben Einstellungen vorgenommen. Wenn eine Einstellung verändert wird, zeigt sich das Ergebnis direkt auf der Optionsseite, denn es wird ein Beispiel für die Sprachsteuerung dargestellt, das aktuell zu jeder Änderung sein Aussehen anpaßt.



Abb. 6.30:
Eigenschaftenfenster

– Vordergrundanzeige

Das Statusfenster der Sprachsteuerung wird standardmäßig immer über allen anderen Fenstern dargestellt. Da dieses Fenster sämtliche wichtigen Informationen zur Spracherkennung darstellt, ist es meist sinnvoll, jederzeit einen Blick darauf werfen zu können.

– Titelleiste anzeigen

Die Titelleiste wird per Voreinstellung angezeigt. Wenn Sie jedoch Platz auf Ihrer Arbeitsoberfläche sparen wollen, können Sie die Anzeige abstellen. Bedenken Sie jedoch, daß Sie dann keinen direkten Zugriff mehr auf die Aktionsknöpfe für Vergrößern, Verkleinern und Schließen haben.

– Menüleiste anzeigen

Bei aktivierter Menüleiste haben Sie direkten Zugriff auf *Aktionen*- und *Hilfe*-Menüs. Wenn die Menüleiste nicht angezeigt wird, können Sie die Menüpunkte aufrufen, indem Sie mit der Maus auf das Systemmenü-Symbol in der Titelleiste klicken. Wenn auch die Titelleiste deaktiviert ist, können Sie mit der rechten Maustaste auf die Sprachsteuerung klicken, um Zugriff auf die Funktionen zu erhalten.

– Knopfleiste anzeigen

Diese Option, die von Anfang an aktiviert ist, zeigt die Schaltflächen für Mikrofon, sprechbare Befehle, Diktierfenster und andere. Das genaue Aussehen und Verhalten dieser Schaltflächen kann noch mit drei weiteren Einstellungen manipuliert werden:

– Kleine Knöpfe anzeigen

Die Schaltflächen werden verkleinert dargestellt, wenn diese Option aktiviert ist. Vorteil ist, daß weniger Platz auf der Arbeitsoberfläche belegt wird.

– Knopftext anzeigen

Bei aktivierter Option wird eine kurze Textbeschreibung in jede Schaltfläche eingefügt. Diese Auswahl empfiehlt sich dann, wenn man noch nicht mit dem Aussehen der Schaltflächen und ihrer Funktion vertraut ist. Allerdings wird dadurch der Platzbedarf der Schaltflächen wesentlich vergrößert.

– Bubble Help anzeigen

Um den Nachteil der Texteinblendung in die Schaltflächen zu vermeiden, können Sie diese Option aktivieren. Wenn Sie nun die Maus auf eine Schaltfläche positionieren, wird nach kurzer Zeit ein kleines Hilfefenster eingeblendet, das die Funktion der Schaltfläche kurz beschreibt.

Auf der zweiten Optionen-Seite werden Einstellungen konfiguriert, die wichtige Informationen während des Diktates zur Verfügung stellen.

– Aussteuerung anzeigen

Diese Anzeige gibt Ihnen die wichtigste Information überhaupt. Sie können mit ihrer Hilfe kontrollieren, daß die von Ihnen gesprochenen Wörter und Befehle weder zu laut noch zu leise

aufgenommen werden. Da in einer falschen Aussteuerung der häufigste Grund für eine fehlerhafte Erkennung liegt, sollten Sie diese Option von Anfang an aktiviert lassen.

– Befehlsprotokoll anzeigen

Bei aktivierter Option, sie ist standardmäßig eingeschaltet, zeigt Ihnen VoiceType den letzten erkannten Befehl an oder eine Nachricht, wenn die letzte akustische Aufnahme nicht erkannt wurde.

– Aktionen im Befehlsprotokoll anzeigen

Wenn Sie diese Auswahl eingeschaltet lassen, wird zusätzlich zum erkannten Befehl auch die zugehörige Aktion mit angezeigt, die durch den Befehl ausgeführt wird.

– Statuszeile anzeigen

Wenn Sie einen zusätzlichen Hinweis zum aktuellen Status des Mikrofons haben möchten, so können Sie diese Aktion aktiviert lassen. Genügt Ihnen jedoch allein das Aussehen des Mikrofon-Schalters, so können Sie diese Option ausschalten, um weiteren Platz auf der Arbeitsoberfläche zu gewinnen.

6.13.3 Sprechbare Befehle

Die dritte Seite der Optionen befaßt sich mit den Einstellungen für das Fenster *Sprechbare Befehle*. Weitere Einstellungen können im Menüpunkt *Aktionen* direkt im *Sprechbare Befehle*-Fenster vorgenommen werden.

Vier Einstellungsmöglichkeiten bietet die dritte Optionsseite der Einstellungen Sprachsteuerung.

– Vordergrundanzeige

Ebenso wie die Sprachsteuerung können Sie das Fenster für *Sprechbare Befehle* so konfigurieren, daß es grundsätzlich über allen anderen Fenstern liegt. Da dieses Fenster für Sie eine Referenz der möglichen Befehle darstellt, sollten Sie immer ohne Aufwand einen Blick darauf werfen können.

– Titelleiste anzeigen

Möchten Sie etwas Platz auf der Arbeitsoberfläche sparen oder diesen Platz für ein paar sprechbare Befehle mehr nutzen, können Sie die Darstellung der Titelleiste unterbinden.

– Menüleiste anzeigen

Dasselbe gilt für die Menüleiste. Um dennoch auf die Funktionen des Menüleiste zugreifen zu können, klicken Sie mit der rechten Maustaste auf die Fensterfläche der *Sprechbaren Befehle*.

– Aktionsknöpfe anzeigen

Am unteren Rand des Fensters werden zwei Schaltflächen, *Trainieren* und *Makro*, angezeigt. Sie dienen dazu, unbekannte Aussprachen für Befehle aufzunehmen und Makros zu erstellen. Möchten Sie diese Schaltflächen nicht sehen, deaktivieren Sie diese Auswahl. Sie können über

den Menüpunkt *Aktionen* dennoch auf die Funktionen dieser beiden Schaltflächen zurückgreifen.

Noch einige weitere Optionen können eingestellt werden, wenn Sie im Fenster *Sprechbare Befehle* auf den Menüpunkt *Aktionen* klicken.

– Programmspezifische Wörter

Mit dieser Option können Sie auswählen, welche Art von Sprachbefehlen angezeigt werden sollen, wenn Sie sich in Anwendungen befinden. Sie können diejenigen Kategorien von Wörtern abwählen, die Sie entweder nicht benutzen wollen oder deren Befehle Sie ohnehin auswendig können. Das *Sprechbare Befehle*-Fenster gibt Ihnen dann genau für das jeweilige Programm, das Sie aktuell verwenden, für alle von Ihnen selektierten Kategorien die möglichen Befehle an.

Tip: Wählen Sie nicht zu viele Kategorien gleichzeitig aus, da ansonsten die Liste aller sprechbaren Befehl so lang werden kann, daß Sie nur noch darauf Zugriff bekommen, wenn Sie mit der vertikalen Schiebeleiste durch alle Befehle blättern.

– Generische Wörter

Sie können an dieser Stelle auswählen, welche Arten von Befehlen, die für alle Programme unter OS/2 Warp 4 gelten, angezeigt werden sollen.

– Wort trainieren

Nach Auswahl dieses Menüpunktes wird der Rekorder für Aussprachen geöffnet. Sie können nun für ein bestimmtes Word die Aussprache aufnehmen und diesen Befehl ab diesem Zeitpunkt verwenden.

– Makro

Wenn Sie diesen Menüpunkt auswählen, wird zunächst ein Auswahlfenster geöffnet, in dem Sie entscheiden müssen, ob das Makro, das Sie im folgenden erstellen wollen, generisch (also für alle Programme gelten) oder programmspezifisch sein soll.

Wenn Sie *generisch* auswählen, werden die Einstellungen der Sprachsteuerung auf der Seite *Steuerelemente* geöffnet. Nach der Auswahl *Programm* hingegen werden die Einstellungen des aktiven Programms auf der Seite *Spracheingabe* angezeigt und Sie können ein Makro nur für dieses eine Programm erstellen.

Genauere Informationen erhalten Sie im Kapitel über die Erstellung von Makros.

– Aussprache editieren

Es wird die Einstellungsseite *Aussprache* in den Eigenschaften der Sprachsteuerung geöffnet. Dort können Sie fehlende Aussprachen hinzufügen oder aufgenommene Aussprachen löschen.

– Sprechbare Befehle – Eigenschaften

Durch diese Auswahl werden die Eigenschaften der Sprachsteuerung auf der dritten *Optionen* Seite geöffnet, wo sämtliche Einstellungen für das *Sprechbare Befehle* Fenster festgelegt werden können.

– Zurück zum Programm

Die Auswahl wird weg vom Fenster *Sprechbare Befehle* zurück zum vorher aktiven Programm bewegt.

– Verlassen

Schließt das Fenster *Sprechbare Befehle*.

6.13.4 Benutzereinstellungen

Wenn Sie die Benutzereinstellungen ändern möchten, ohne dazu das Icon auf der Arbeitsoberfläche zu verwenden, so können Sie dies tun, indem Sie die Einstellungen der Sprachsteuerung aufrufen und dort die Seite *Benutzer* anwählen. Auf dieser Seite werden die aktuellen Einstellungen zum Benutzer wie Name und Beschreibung der Registrierung sowie der verwendete Wortschatz angezeigt. Zusätzlich kann man auswählen, ob dazu noch die verwendete Sprache in der Zeile zur Beschreibung der Registrierung angezeigt werden soll. Diese Option ist vor allem dann sinnvoll, wenn mehrere Sprachen auf einem System installiert sind.

Wenn Sie nun den aktuellen Benutzer ändern möchten, wählen Sie aus der Liste *Name* den gewünschten Benutzer aus und klicken Sie auf *Anwenden*. Falls Sie einen neuen Benutzer anlegen möchten, klicken Sie auf *Neu* und geben Sie die Benutzerkennung ein, die den neuen Benutzer identifizieren soll. Gleichzeitig wird auf der Arbeitsoberfläche ein Symbol für den neuen Benutzer erstellt.

6.13.5 Audioeinstellungen

Im Notizbuch für die Einstellungen der Sprachsteuerung gibt es unter der Bezeichnung *Audio* drei Seiten, auf denen sämtliche Einstellungen für die Aufnahme und Wiedergabe konfiguriert werden können.

Wenn Sie Änderungen auf diesen Seiten vornehmen, werden diese dann aktiv, wenn Sie das Notizbuch geschlossen und das Mikrofon einmal aus- und dann wieder eingeschaltet haben.

Auf der ersten Seite können Sie alle Einstellungen, die mit der Hardware zu tun haben, einstellen.

– Adapterkarte

Eine Liste aller in Ihrem PC installierten Soundkarten wird angezeigt. In der Regel wird sich nur eine solche Karte im PC befinden und deshalb müssen Sie normalerweise hier keine Änderung vornehmen.

– Eingangsanschluß

Sie können wählen, an welchem Eingang Ihrer Soundkarte das Mikrofon angeschlossen wird. Mikrofone müssen in der Regel am Eingang *Mikrofon* angeschlossen sein. Wird jedoch das Signal durch einen externen Mikrofonvorverstärker an die Soundkarte weitergeleitet, kann es unter Umständen notwendig sein, den Anschluß *Eingang* zu wählen. Achten Sie bei der Aus-

wahl des Eingangs auch darauf, daß Ihr Mikrofon auch physisch an dem Eingang angeschlossen ist, den Sie in dieser Einstellung auswählen!

– Eingangsspannung

Mit diesem Regler können Sie einstellen, wie stark das anliegende Signal, das vom Mikrofon kommt, verstärkt werden soll. Stellen Sie den Regler so ein, daß die Aussteuerungsanzeige immer im grünen Bereich liegt. Nur dann kann VoiceType optimal Ihre Sprache erkennen.

– Ausgangsspannung

Mit diesem Regler stellen Sie die Wiedergabelautstärke Ihres Lautsprechers ein.

– Widerrufen

Mit dieser Schaltfläche können Sie sämtliche Änderungen rückgangig machen, die Sie seit dem Öffnen des Fensters vorgenommen haben.

– Standard

Wenn Sie auf diese Schaltfläche klicken, werden die Voreinstellungen für diese Seite wiederhergestellt.

Die zweite Seite beschäftigt sich mit der Art und Weise, wie genau VoiceType »hinhört«. Der Regler befindet sich standardmäßig in der Mitte zwischen *Annähernd* und *Exakt*. Wenn Sie den Regler nach links schieben, wird Ihre Aussprache weniger kritisch untersucht, das bedeutet, daß unter Umständen mehr Befehle erkannt werden könnten. Gleichzeitig können allerdings andere Geräusche wie das Klingeln eines Telefons oder Husten als ein Befehl interpretiert werden. Genau das Gegenteil Sie, wenn Sie den Regler nach rechts bewegen. Die Voreinstellung ist in der Regel für die meisten Fälle die beste Wahl, verändern Sie diesen Regler also nur, wenn Sie einen triftigen Grund dafür haben.

Zusätzlich können Sie noch die Priorität auf eine Funktion legen, die VoiceType besonders beachten soll:

Geschwindigkeit: VoiceType legt besonderen Wert auf schnelle Erkennung, auch wenn dadurch die Erkennung weniger genau wird. Diese Einstellung ist vor allem für langsamere Systeme sinnvoll.

Präzision: Auch wenn die Erkennungsgeschwindigkeit sinkt, so soll VoiceType bei Auswahl dieser Option die Wörter möglichst genau analysieren und die optimale Erkennung bringen.

Gleich: Es wird ein Kompromiß eingegangen zwischen Geschwindigkeit und Erkennungsgenauigkeit. Dies ist die Voreinstellung und für die meisten Computer optimal.

Die dritte Seite läßt Einstellungen für die kontinuierliche Sprache bei der Navigation zu. Sie können festlegen, wie lange eine Pause innerhalb eines Befehls sein darf, der als ein kontinuierliches Kommando gesprochen werden soll. Sie können zwischen 0 und 10 Sekunden wählen.

Am Ende eines Befehlssatzes wartet VoiceType zwischen 0 und maximal 3 Sekunden, bis es den nächsten Befehl erwartet.

6.13.6 Hervorhebung

In der folgenden Einstellungsseite können Sie festlegen, ob der Rahmen zur Markierung des aktiven Objekts eingeschaltet sein soll und wie er aussehen soll. Sie können sowohl die Farbe als auch die Dicke des Rahmens festlegen.

Tip: Die Funktionalität des Rahmens benötigt relativ viel Rechenleistung. Wenn Sie nach einer gewissen Eingewöhnungsphase mit dem Rahmen wissen, wie VoiceType arbeitet, können Sie den Rahmen ausschalten, um etwas mehr Rechenleistung zu nutzen.

6.13.7 Ruhezustand

Der Ruhezustand ist immer dann sinnvoll, wenn Sie für einige Zeit VoiceType nicht nutzen werden, aber trotzdem jederzeit durch ein Sprachkommando die Spracherkennung wieder aktivieren wollen. Sie könnten zwar auch das Mikrofon einfach eingeschaltet lassen, doch dies erfordert mehr Rechenleistung.

Auf der Einstellungsseite für den Ruhezustand können Sie festlegen, ob der Ruhezustand automatisch aktiviert werden soll, sobald VoiceType gestartet wird. Zusätzlich können Sie bestimmen, ob und nach welcher Zeit oder nach wie vielen falsch erkannten Wörtern VoiceType automatisch in den Ruhezustand wechseln soll.

6.13.8 Steuerelemente

Auf der Seite *Steuerelemente* können Sie sämtliche Befehle für die Steuerung, geordnet nach ihren Funktionen, verändern, trainieren oder neue erstellen.

Wenn Sie zum Beispiel einen Befehl erstellen wollen, der immer und überall ausgesprochen werden kann, so wählen Sie die Kategorie *Globale Wörter* und klicken Sie auf *Erstellen*. Danach können Sie bestimmen, welche Funktionen durch den Befehl ausgeführt werden sollen.

Sie können auch über *Wort trainieren* für einen Befehl eine völlig neue Aussprache aufnehmen oder eine Aussprache hinzufügen, so daß danach der Befehl durch mehrere unterschiedliche Sprachkommandos aktiviert werden kann.

So könnten Sie *Ruhezustand* durch *Geh Schlafen* ersetzen oder *Ruhezustand Ende* und *Wach auf* als zwei mögliche Aussprachen definieren, die das Mikrofon einschalten sollen.

6.13.9 Aussprache ändern oder hinzufügen

Die Seite *Aussprachen* ist abhängig von dem gerade aktiven VoiceType Benutzer, denn jeder Benutzer wird mit der Zeit andere Befehle trainieren als ein anderer. Auf dieser Seite können Sie Aussprachen für Wörter aufnehmen, die VoiceType zwar als Befehle identifizieren kann, jedoch kein Sprachmuster für das Wort hinterlegt ist.

Sobald Sie die Aussprache für ein Wort aufgenommen haben, das Sie aus der Liste der *Fehlenden Wörter* ausgewählt haben, wird dieses Wort in die Liste der *Aufgenommenen Wörter* übertragen. In der Liste der *Aufgenommenen Wörter* stehen all diejenigen Wörter, die Sie in Ihr persönliches Sprachvokabular aufgenommen haben und die nur Ihnen zur Verfügung stehen.

6.14 Makros und Vorlagen

In diesem Kapitel werden die verschiedenen Arten von Makros etwas genauer betrachtet. Dabei kommen sowohl Diktiermakros als auch Steuerungs- und Anwendungsmakros zur Sprache.

6.14.1 Diktiermakros

Wenn Sie bestimmte Textbausteine oder Floskeln immer wieder verwenden, denkbar wären ein Briefkopf oder eine Grußformel, so können Sie sich ein Diktiermakro erstellen und mit nur einem Kommando den gesamten Textteil auf den Bildschirm bringen. Sie können auch ganze Formulare entwickeln, in denen mehrere Felder enthalten sind, die Sie per Diktat ausfüllen können.

Zur Erstellung solcher Makros oder Vorlagen wird der Diktiermakro-Editor verwendet. Die mit ihm erstellten Textbausteine können in jeder Anwendung verwendet werden, die direktes Diktieren zulassen. Sie haben also die Möglichkeit, die Anzahl der Makros, die mit VoiceType mitgeliefert werden, selbst zu vergrößern.

Makros erstellen und ändern

Um ein Makro zu erstellen, klicken Sie in der Menüzeile des Diktiermakro-Editors auf *Editieren* und dann auf *Makro erstellen*. Sie erhalten den Eingabeschirm für Makros. Zunächst müssen Sie einen Makronamen eingeben, mit dem Sie später während des Diktierens das Makro aufrufen wollen.

Tip: Verwenden Sie für den Makronamen niemals ein einzelnes Wort, sondern immer eine Kombination aus zwei oder drei Wörtern. Wenn Sie das Makro mit nur einem Wort benennen und dieses diktieren, hat VoiceType ein Problem zu entscheiden, ob Sie nun das Wort oder das Makro meinen.

Sinnvoll ist auch, eine Beschreibung für das Makro einzugeben, so können Sie später jederzeit verifizieren, welche Funktion das Makro hat und ob Sie es auch weiterhin benötigen.

Bei der Eingabe des Makrotextes können Sie auf einige Sonderfunktionen zugreifen. So können Sie das aktuelle Datum und die Uhrzeit automatisch einfügen lassen oder sogar externe ReXX-Scripts oder DLLs oder Systemaufrufe direkt aus dem Makro heraus starten.

Nachdem Sie das Erfassen des Makros beendet haben, wird es in die Liste der verfügbaren Makros aufgenommen.

Sie können ein Makro später noch verändern, indem Sie es aus dieser Liste auswählen und mit einem Doppelklick öffnen.



Abb. 6.31:
Makros erstellen

Vorlagen erstellen

Vorlagen sind eine besondere Form von Makros. Sie enthalten mehrere Felder, in denen durch Diktieren Text erfaßt werden kann. Sie können von einem Texteingabefeld zu einem anderen durch *Nächstes Feld* und *Vorheriges Feld* navigieren.

Sie erstellen Vorlagen im Diktiermakro-Editor durch *Editieren|Vorlage erstellen*. Sie können nun Textteile eingeben und Felder hinzufügen, in die diktiert wird. Ebenso wie bei Makros können Sie innerhalb von Vorlagen externe Routinen oder andere Makros aufrufen.

Unter *Optionen* können Sie die Position des Makros oder der Vorlage im Text vordefinieren.

Aussprache hinzufügen

Wenn VoiceType keine Aussprache zu Ihrem neuen Makro kennt, erscheint das Fenster zur Erfassung der Akustik. Wenn Ihr Makroname aus mehreren Wörtern besteht, klicken Sie auf *erweitert* und geben Sie an, daß sie den Befehl wie ein Wort aussprechen werden. Dann können Sie die Akustik aufnehmen.

Der Diktiermakro-Editor zeigt alle verfügbaren Makros in einer Liste an. Falls in der ersten Spalte der Liste ein rotes X vor einem Makro angezeigt wird, müssen Sie noch die Akustik für Ihren Makrobefehl trainieren, wenn Sie das Makro verwenden wollen. Wählen Sie das Makro aus der Liste aus, und klicken Sie in der Menüzeile des Editors auf *Aussprache* Trainieren, um auf den Akustik-Rekorder zu gelangen. Alles Weitere erfolgt analog zur soeben beschriebenen Aufnahme der Akustik.

Im- und Export von Makros

Sie können Makros im- und exportieren, um sie zum Beispiel von einem Computer zu einem anderen zu übertragen. Bei dieser Übertragung geht Ihnen allerdings die akustische Information verloren, falls Sie diese für das Makro explizit trainieren mußten.

Um ein Makro zu importieren klicken Sie auf *Datei|Import* und wählen im Dateiauswahlfenster die Datei mit dem gewünschten Makro aus.

Der Export eines Makros funktioniert analog. Nachdem Sie das zu exportierende Makro mit der Maus ausgewählt haben, klicken Sie auf *Datei|Export* und geben Sie einen Dateinamen ein. Danach wird das Makro in diese Datei exportiert.

6.14.2 Steuerungsmakros

Mit Steuerungsmakros können Sie unter OS/2 Warp 4 sämtliche sogenannte »Steuerelemente« (Controls) steuern. Dies sind gewisse Objekte des Betriebssystems, so zum Beispiel Schiebeleisten oder Auswahllisten. Sie können sich auf der Seite *Steuerelemente*des Einstellungsnotizbuches für die Sprachsteuerung eine komplette Liste der Controls von OS/2 Warp 4 ansehen.

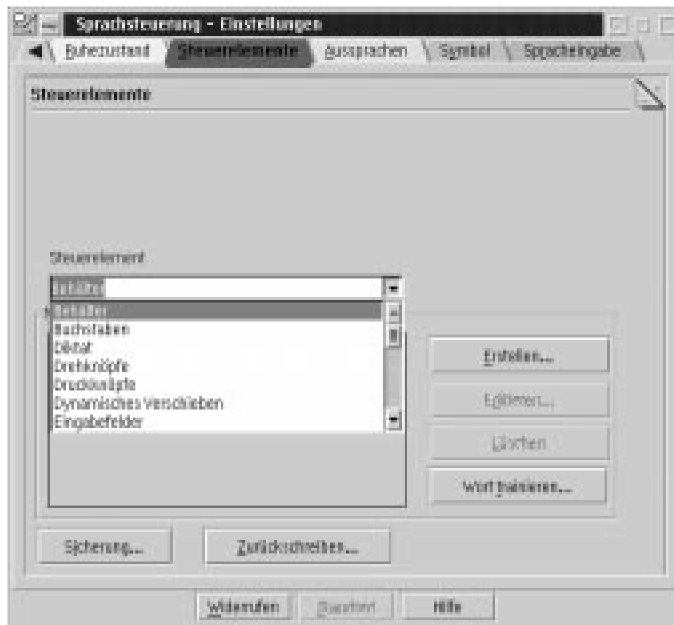


Abb. 6.32: Die Controls von OS/2

Für jede Kategorie gibt es bereits vorgefertigte Makros, und Sie können noch beliebig viele Makros selbst hinzufügen. Dabei müssen Sie jedoch darauf achten, daß in einer Kategorie jeder Makroname nur einmal vorkommen darf. In zwei oder mehr unterschiedlichen Katego-

rien darf natürlich derselbe Name verwendet werden, so etwa der Befehl »ab« bei den Schiebleisten und bei den Auswahllisten.

Steuerungsmakros erstellen

Wenn Sie ein Steuerungsmakro erstellen wollen, öffnen Sie das *Eigenschaften*-Fenster der Sprachsteuerung und wählen Sie die Seite *Steuerelemente* aus. Dort können Sie entscheiden, für welche Art von Controls das neue Makro gelten soll. Danach klicken Sie auf *Erstellen* und geben Sie den Namen und die Beschreibung des Makros ein. Die Beschreibung ist deshalb wichtig, da Sie anhand dieser auch nach einiger Zeit noch auf den ersten Blick erkennen können, welche Funktion das Makro hat, ohne daß Sie dieses Makro durch *Editieren* erneut öffnen müssen.

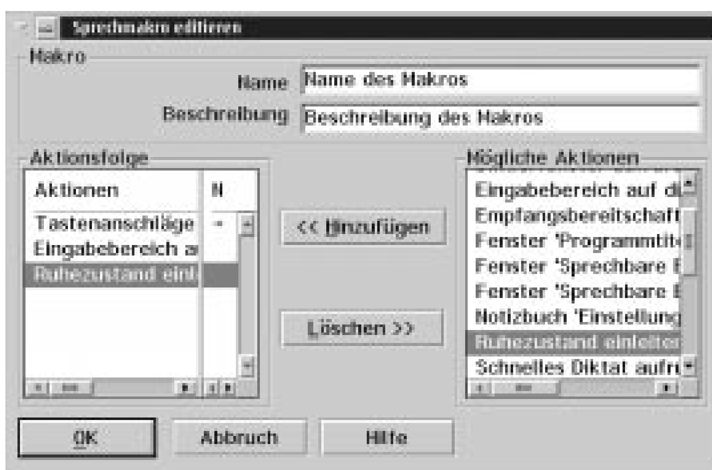


Abb. 6.33: Makroerstellung

Im unteren Teil des Fensters wird auf der rechten Seite die Liste aller möglichen Aktionen angezeigt, links kann die Aktionsfolge Ihres Makros betrachtet werden. Wählen Sie auf der rechten Seite eine Aktion aus und klicken Sie auf *Hinzufügen*, um die Aktion auf die linke Seite zu kopieren. Wenn Sie eine falsche Aktion hinzugefügt haben, können Sie diese mit *Entfernen* wieder aus dem Makro verbannen. Wichtig zu wissen ist, daß Sie keine Aktionen nachträglich einfügen können! Jede hinzugefügte Aktion wird an das Ende des Makros angehängt. Deshalb sollten Sie die Aktionsfolge Ihres Makros am besten »von Hand« durchspielen und aufschreiben, bevor Sie das Erfassen des Makros beginnen. Dies gilt vor allem dann, wenn Sie auch Tastenanschläge erfassen wollen. Beachten Sie in diesem Fall auch das Kapitel »Tastenanschläge erfassen«.

Wenn Sie die Befehlsfolge vollständig erfaßt haben, klicken Sie auf *Ok*. Erkennt VoiceType die Aussprache Ihres Makros nicht, wird automatisch eine Nachricht angezeigt, in der Sie entscheiden können, ob Sie die Aussprache trainieren wollen. Wenn ja, wird der Rekorder für die Aussprache geöffnet und Sie können die Akustik trainieren. Falls nein, müssen Sie den Befehl später trainieren, sonst können Sie Ihr eigenes Makro nicht verwenden.

Sobald die Akustik trainiert ist, steht Ihnen die Funktionalität Ihres Makros zur Verfügung.

Wenn Sie ein Makro nachträglich verändern möchten, wählen Sie das entsprechende Makro aus und klicken Sie auf *Editieren*. Sie gelangen danach wieder zu dem Fenster, in dem Sie die Aktionen des Makros erfassen oder löschen können.

Zum Löschen eines gesamten Makros wählen Sie das Makro aus und klicken Sie auf *Löschen*.

Steuerungsmakros sichern und wiederherstellen

Wenn Sie Ihre Makros auf andere Computer übertragen wollen, können Sie Ihre Makros in eine Datei sichern und später wiederherstellen. Beim Sichern wird jeweils ein Makro in eine Datei mit der Erweiterung ».SPM« gesichert. Sie können diese Datei auf eine Diskette speichern und so auf einen anderen Computer übertragen, um auch dort nach dem Wiederherstellen des Makros aus dieser Datei auf Ihr Makro zugreifen zu können.

Steuerungsmakros, die auf einem System erstellt wurden, sind für alle Benutzer verfügbar, die sich auf diesem System befinden. Allerdings muß jeder Benutzer die Aussprache für das Makro trainieren, wenn VoiceType die Akustik nicht schon von sich aus kennt.

6.14.3 Anwendungsmakros

Es gibt bereits bei der Installation von VoiceType eine ganze Reihe von Makros, die für alle OS/2-Anwendungen funktionsfähig sind. Wenn Sie jedoch zusätzliche Funktionen für ein spezielles Programm mit einigen Makros automatisieren möchten, so können Sie Anwendungsmakros erfassen.

Diese Art von Makros sind vor allem für Anwendungen nützlich, die im Win-OS/2 oder DOS-Fenster ablaufen. Makros funktionieren nicht in den jeweiligen Gesamtbildschirmen!

Anwendungsmakros erstellen

Da Anwendungsmakros nur speziell für ein bestimmtes Programm gelten, werden sie auch in den Eigenschaften des jeweiligen Programms auf der Seite *Spracheingabe* erfaßt. Achten Sie dabei darauf, daß es sich tatsächlich um ein Programmobjekt und nicht umeinen Ordner handelt. Für Ordner können keine Anwendungsmakros erfaßt werden.

Klicken Sie auf *Erstellen* und geben Sie den Namen des Makros und eine Beschreibung ein. Alle weiteren Aktionen zum Erfassen des Makros gehen genauso vonstatten, wie unter *Steuerungsmakros erstellen* beschrieben.

Anwendungsmakros sichern und wiederherstellen

Auch das Sichern und Wiederherstellen von Anwendungsmakros erfolgt auf dieselbe Art und Weise wie bei den Steuerungsmakros. Nach dem Klick auf *Sichern* wird eine .SPM Datei erstellt, die dann auf andere Computer übertragen werden können. Bedenken Sie jedoch, daß sich das Format der Anwendungsmakros von dem der Steuerungsmakros unterscheidet. Sie können also nicht eine Sicherung der einen Kategorie in der anderen Makroart importieren.

In der so generierten Datei werden alle Makros gespeichert, die Sie für dieses spezielle Programm erstellt haben.

6.14.4 Tastenanschläge erfassen

Zur Erstellung von eigenen Makros ist es möglich, Tastenanschläge zu erfassen, um diese während der Laufzeit des Makros an das Zielobjekt zu senden. Diese Tastenanschlägen können Buchstaben, Sonderzeichen, Funktionstasten und spezielle Tasten wie [Alt], [Strg] oder [Enter] sein.



Abb. 6.34: Erfassen von Tastenanschlägen

Die Funktion zum Erfassen von Tastenanschlägen ist für Anwendungsmakros und für einige Steuerungsmakros verfügbar (etwa für globale Wörter). Die Funktion *Tastenanschläge erfassen* soll anhand eines Beispiels für ein global gültiges Makro erklärt werden.

Es soll ein Makro zum automatisierten Systemabschluß per Sprache erstellt werden. Öffnen Sie dazu die Eigenschaften der Sprachsteuerung und wählen Sie die Seite *Controls*. Selektieren Sie *Globale Wörter* und klicken Sie auf *Erstellen*. Als Name geben Sie *Systemabschluß durchführen* ein und als Beschreibung zum Beispiel *OS/2 herunterfahren*. Wählen Sie nun aus den möglichen Aktionen die Aktion *Tastenanschläge senden* aus und klicken Sie auf hinzufügen. Die Aktion wird nach links übernommen und ein Fenster erscheint.

In diesem Fenster können alle Tastenanschläge erfaßt werden. Wenn Sie keine Sondertasten erfassen wollen, können Sie direkt in das Fenster die Tastenanschläge eingeben. Möchten Sie jedoch Sondertasten wie [Enter] oder [Alt] erfassen, klicken Sie in der Zeile *Alle Tastenanschläge erfassen* auf *Start*. Nun können Sie so lange Tasten erfassen, bis Sie *Stop* angeklickt haben. Beachten Sie, daß auch die Löschen-Taste aufgezeichnet wird. Sie können also in diesem Modus keine Korrekturen durchführen. Erst wenn *Stop* gedrückt wurde, können Sie Eingabefehler beheben. Auch Kombinationen von Tasten (etwa [Alt] und [F10]) können erfaßt werden. Drücken Sie dazu die erste Taste und halten Sie diese gedrückt. Tippen Sie danach auf die zweite und alle weiteren Tasten, bis die Kombination vollständig erfaßt ist.

Sondertasten werden durch eckige Klammern eingeschlossen, zum Beispiel [Alt]. Wenn mehrere Tasten gleichzeitig gedrückt werden, so sind die eckigen Klammern geschachtelt,

[Alt][F10] bedeutet somit, daß die Taste [Alt] gedrückt wurde und zusätzlich [F10]. [Alt][F10] bedeutet hingegen, daß die beiden Tasten nacheinander kurz angetippt wurden. Dieses Wissen ist dann wichtig, wenn Sie Tastenkombinationen erfassen wollen, die unter OS/2 eine sehr hohe Priorität haben wie zum Beispiel [Strg]+[Esc] zum Aufrufen der Fensterliste.

Tip: Um ein Makro möglichst übersichtlich und editierbar zu gestalten, geben Sie nicht zu viele Tastenanschläge in ein Eingabefenster ein. Auch wenn Ihr Makro beim Testen nicht fehlerfrei ausgeführt wird, erleichtert Ihnen diese Technik, den Fehler leichter zu finden und zu beheben.

Zum Erfassen des Makros für den Systemabschluß können wir nun die Tastenanschläge erfassen:

Zuerst müssen wir die Fensterliste aufrufen. Wir klicken also auf *Start* in der Zeile *Alle Tastenanschläge erfassen* und tippen zuerst [Strg] und danach [Esc]. Hätten wir die beiden Tasten wie gewohnt mit [Strg]+[Esc] gedrückt, wäre lediglich die Fensterliste erschienen, aber es wären keine Tastenanschläge erfaßt worden. Deshalb klicken wir jetzt auf *Stop* und editieren den Text von Hand. Wir löschen einfach das erste *]* und setzen am Ende der Zeile dafür ein neues *]*, so daß nun im Fenster [Strg][Esc]] steht. Danach klicken wir auf *Ok*, um das Fenster zu schließen.

Es folgt ein zweiter Tastenanschlag, in dessen Fenster wir »A« für das Auswählen der Arbeitsoberfläche eingeben. *Ok* schließt das Fenster.

Zum Umschalten auf die Arbeitsoberfläche fehlt nun noch das Erfassen der [Enter]-Taste.

Nun muß alles auf der Arbeitsoberfläche deselektiert werden. Wir erreichen dies durch die Tastenkombination [Strg][AltGr][ß].

Der Aufruf des Kontextmenüs erfolgt durch [Shift][F10], natürlich ebenfalls wieder in einem eigenen Fenster für Tastenanschläge erfaßt.

Um das Wort Systemabschluß auszuwählen, müssen wir einen Tastenanschlag »y« senden.

Nun fehlt nur noch die letzte Taste [Enter], um den Systemabschluß zu aktivieren.

Mit dem Klick auf *Ok* wird das Makro abgespeichert und falls VoiceType die Aussprache noch nicht kennt, können Sie nun den Befehl trainieren. Ab diesem Zeitpunkt können Sie Ihr OS/2 Warp 4 auch per Sprache beenden. Viel Spaß!

Wenn Sie später in einem Makro die erfaßten Tastenanschläge verändern wollen, klicken Sie auf *Editieren* und dann rechts neben dem *Tastenanschläge erfassen* auf den Pfeil. Dadurch wird das Fenster mit den eingegebene Tastenanschlägen zum Editieren geöffnet.

6.14.5 Sprungbefehl-Makros

Diese Art von Makros dienen dazu, ein Programm, einen Ordner oder ein Objekt zu öffnen. Einige Makros dieser Art sind bereits in VoiceType integriert, so zum Beispiel für einige häufig benutzte Ordner der Arbeitsoberfläche.

Sie finden für jedes Objekt in den Einstellungen unter *Spracheingabe* eine Seite, auf der Sie solch ein Makro erstellen können.

Programme haben also unter dem Eintrag *Spracheingabe* zwei Seiten, auf denen Makros erstellt werden können. Auf der ersten Seite werden die Anwendungsmakros aufgelistet und auf der zweiten die Sprungbefehl-Makros zum Start des Programms. Ordner besitzen nur eine Seite für Sprungbefehl-Makros.

Die Erstellung eines Sprungbefehl-Makros erfolgt, indem Sie auf *Erstellen* klicken und einen Namen sowie die Beschreibung für das Makro eingeben. Zuletzt können Sie noch unter *Aktion* auswählen, auf welche Art das Objekt, der Ordner oder das Programm geöffnet werden soll.

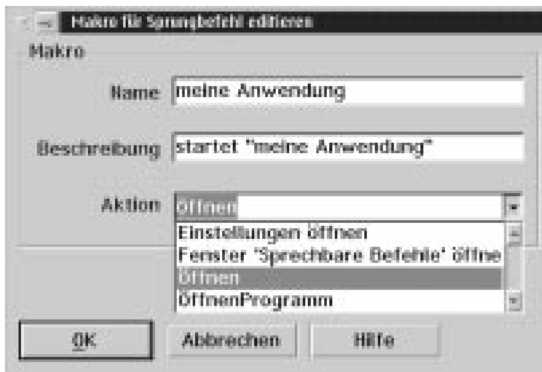


Abb. 6.35:
Sprungbefehl-Makros

6.14.6 Makronamen trainieren

Um einen Makronamen zu trainieren, müssen Sie zunächst entscheiden, um welches Makro es sich handelt.

– Steuerungsmakros können Sie trainieren, indem Sie die Eigenschaften der Sprachsteuerung öffnen und die *Controls*-Seite auswählen. Wählen Sie das zu trainierende Makro aus, und klicken Sie auf *Wort trainieren*.

– Anwendungsmakros traininerен Sie in der Seite *Sprache* des Eigenschaften-Notizbuchs für das entsprechende Programm. Dort können Sie mit *Wort trainieren* die Akustik aufnehmen.

– Voice-Sprungbefehl-Makros trainieren Sie im Eigenschaften-Notizbuch des Ordners, für den das Makro gilt. Wählen Sie die Seite *Spracheingabe* aus, und klicken Sie auf *Wort trainieren*.

– Diktiermakros können nur im Diktiermakro-Editor trainiert werden.

6.15 Tips und Performance-Tuning

In diesem Kapitel sollen abschließend einige Tips gegeben werden, wenn VoiceType nicht so reagiert, wie Sie es vielleicht erwarten würden. Diese Tips können Ihnen vielleicht helfen, den Grund für die ungewöhnliche Reaktion der Spracherkennung zu finden und das Problem zu beseitigen.

Außerdem wird auf das Thema der Geschwindigkeitsverbesserung eingegangen, und wie Sie mit möglichst wenig Aufwand mehr aus Ihrem VoiceType herausholen können. Zum Schluß wird noch kurz auf die Erweiterung der Spracherkennungsfunktion durch ein spezielles Update für Fachwortschätze eingegangen.

VoiceType reagiert nicht

Es kann vorkommen, daß VoiceType überhaupt nicht mehr reagiert. Prüfen Sie in diesem Fall zuerst, ob Sie im Steuerzentrum weitere Informationen erhalten. Sehen Sie sich die Mikrofontaste an und prüfen Sie, ob das Mikrofon vielleicht nur ausgeschaltet ist. Es kann auch sein, daß ein Symbol auf der Mikrofontaste zu sehen ist, das einem »Halteverbotsschild« ähnelt.

Ist dies der Fall, so ist ein schwerer interner Fehler aufgetreten und die gesamte Spracherkennungsfunktion wurde deaktiviert. Schließen Sie das Steuerzentrum und starten Sie VoiceType erneut. Läßt sich dadurch das Problem nicht beheben, so führen Sie einen Systemabschluß durch und starten Sie Ihren Computer neu.

VoiceType erkennt Wörter nicht

Wenn Sie immer wieder beobachten müssen, daß VoiceType regelmäßig Wörter nicht erkennt, so haben Sie einige Möglichkeiten zu prüfen, woran die fehlerhafte Erkennung liegen kann.

- Überprüfen Sie, ob das Mikrofon korrekt positioniert ist. Es sollte weder zu weit noch zu kurz vom Mund entfernt sein, denn sonst wird Ihre Stimme nicht korrekt aufgezeichnet. Achten Sie darauf, daß sich die Membran des Mikrofons nicht direkt vor Ihrem Mund befindet, da die Luft, die beim Sprechen aus dem Mund strömt, Windgeräusche im Mikrofon erzeugt.
- Wenn es sich um einen nicht erkannten Befehl handelt, trainieren Sie die Aussprache des Befehls. Wählen Sie dazu den Befehl im *Sprechbare Befehle*-Fenster aus und klicken Sie auf *Trainieren* oder öffnen Sie die Einstellungen der Steuerzentrale auf der Seite *Aussprache* und wählen Sie dort das fehlende Wort zum Training aus.
- Wenn ein Wort während des Diktierens nicht erkannt wird, korrigieren Sie den Befehl, wie im Kapitel »Diktieren mit VoiceType« beschrieben.
- Überprüfen Sie die Audio-Einstellungen. Sie finden diese auf der Seite *Audio* im Einstellungs-Notizbuch des Steuerzentrums. Dort können Sie den Eingangspegel für Ihr Mikrofon einstellen und die Erkennungsgenauigkeit von VoiceType justieren.

Keine Wiedergabe durch den Lautsprecher

Wenn Sie beim Wiedergeben eines Diktates nichts hören, starten Sie zunächst das Programm *Installation prüfen* und lassen Sie sich dort die Nachricht wiedergeben. Wenn auch diese Nachricht nicht hörbar ist, überprüfen Sie den Lautstärkeregler am Lautsprecher. Kontrollieren Sie die physische Verbindung zwischen Ihrer Soundkarte im PC und dem Lautsprecher. Wenn Sie aktive Lautsprecher verwenden, überprüfen Sie die Stromversorgung.

Sie können auch im Programm *Installation prüfen* eine Nachricht selbst aufnehmen und wiedergeben lassen, um den Lautsprecher und das Mikrofon zu überprüfen.

Registrierung bricht ab

Wenn die Berechnung Ihrer Registrierung vorzeitig abbricht, wird eine Nachricht angezeigt, die Sie auf eine README-Datei verweist. Dort finden Sie weitere Informationen, wie Sie den Fehler beheben können.

Nach dem Abbruch kann eine der folgenden Nachrichten angezeigt werden:

– **Nicht genügend Speicher**

Auf Ihrer Festplatte oder auf der ausgewählten Partition der Festplatte ist nicht genügend Speicherplatz vorhanden.

Unterbrechen Sie die Registrierung und schaffen Sie mehr Platz auf der Festplatte oder Partition. Wenn Sie die Registrierung wieder aufnehmen, wird Ihnen der verfügbare Plattenplatz angezeigt.

– **Fehler beim Schließen einer Datei**

Auch diese Meldung weist auf zu wenig Speicherplatz auf der Festplatte hin. Unterbrechen Sie die Registrierung und sorgen Sie für mehr Platz. Nehmen Sie danach die Berechnung wieder auf.

– **Fehlerhafte Daten im Verzeichnis /SPCH_RW**

Eine Datei im Verzeichnis \SPCH_RW wurde zerstört.

Unterbrechen Sie die Berechnung und nehmen Sie diese wieder auf.

– **Lesefehler im Verzeichnis /SPCH_RW**

Eine Datei wurde nicht im Verzeichnis \SPCH_RW gefunden oder sie ist beschädigt.

Unterbrechen Sie die Berechnung und nehmen Sie diese wieder auf.

– **Schreibfehler im Verzeichnis /SPCH_RW**

Das Registrierungsprogramm konnte keine Dateien auf das Ziellaufwerk schreiben. Es könnte sein, daß die Festplatte inzwischen keinen Platz mehr frei hat.

Nach dem Aussetzen der Registrierung schaffen Sie Platz auf der Festplatte, indem Sie Dateien auf ein anderes Laufwerk verschieben oder Dateien löschen. Beim Wiederaufnehmen wird Ihnen die verfügbare Speicherkapazität angezeigt.

Mikrofon schaltet sich aus

Es kann vorkommen, daß sich während des Diktierens plötzlich das Mikrofon abschaltet und eine Fehlernachricht angezeigt wird, die meldet, daß der Audio-Speicher voll sei.

In diesem Fall sind Sie mit Ihrem Diktat der Textausgabe von VoiceType zu weit vorausgeeilt und Sie können so lange nicht weiterdiktieren, bis VoiceType den Audio-Speicher geleert hat. Sobald dies geschehen ist, können Sie das Mikrofon wieder anschalten und Ihr Diktat fortsetzen.

Sie können diesen Fehler vermeiden, indem Sie für mehr verfügbaren Arbeitsspeicher sorgen. Schließen Sie einige geöffnete, aber aktuell nicht benötigte Programme.

Der Fehler kann auch dann auftreten, wenn Sie das Diktat gestartet haben, aber für längere Zeit (etwa 100 Sekunden) nichts sprechen. Der Audio-Speicher kann in etwa 100 Sekunden aufnehmen. Durch das gestartete Diktat wird der Speicher gefüllt, er kann aber nicht abgearbeitet werden, da Sie ja kein Wort diktiert haben.

Sie können den Fehler beheben, indem Sie einfach das Mikrofon wieder einschalten und das Diktat starten.

6.16 Performance optimieren

Auch wenn Ihr Rechner die Anforderungen für VoiceType erfüllt, so kann es doch vorkommen, daß Ihnen subjektiv die Antwortzeit Ihres Systems als zu hoch erscheint. Hier sollen nun ein paar Tips folgen, wie Sie die Arbeitsgeschwindigkeit erhöhen können. Dabei sollen vor allem die Gesichtspunkte Hauptspeicher, Prozessor, Festplatte und Anwendungsprogramme betrachtet werden.

Hauptspeicher

Klar, Arbeitsspeicher kann nur ersetzt werden durch noch mehr Arbeitsspeicher. Auch VoiceType profitiert von jedem Megabyte mehr, das zur Verfügung steht. Erweitert man den Hauptspeicher von den minimal geforderten 24 Mbyte auf 32 Mbyte oder gar 40 Mbyte, so kann man einen deutlichen Sprung in der Geschwindigkeit erkennen. Allerdings läßt sich dieser Trend nicht linear fortschreiben. Eine Erweiterung des Speichers von 64 Mbyte auf 96 Mbyte wird bei der Arbeitsgeschwindigkeit von VoiceType so gut wie gar nicht mehr ins Gewicht fallen.

Sie sollten bei Ihrer Kalkulation für die Größe Ihres Arbeitsspeichers jedoch auch die Anwendungen berücksichtigen, die Sie zusammen mit VoiceType verwenden wollen. Wenn Sie in ein Programm diktieren wollen, das alleine schon 16 Mbyte RAM belegt, werden Sie mit einer Speichergröße von 24 Mbyte auf keinen grünen Zweig kommen. Sie sollten Ihre Speichergröße mit folgenden Minimalwerten kalkulieren:

OS/2 Warp	8 Mbyte
VoiceType	16 Mbyte
mit VT benutzte Anwendungen	x Mbyte
Minimales RAM	24 + x Mbyte

In den Newsgroups zum Thema OS/2 Warp 4 Beta wurde oft diskutiert, daß ein möglichst schneller und großer Cache-Speicher die Geschwindigkeit von VoiceType erhöht. Ich selbst habe dazu keine speziellen Tests durchgeführt, allerdings wirkt sich der Cache-Speicher grundsätzlich positiv auf das gesamte OS/2 System aus. Deshalb gehe ich davon aus, daß diese Speicherart vielmehr eine allgemeine Performance-Steigerung bringt als nur speziell für VoiceType.

Prozessor

VoiceType macht intensiven Gebrauch von der Gleitkommaeinheit des Prozessors. Frühere Versionen von VoiceType konnten auf einen Spezialprozessor zurückgreifen, der speziell im Anwendungsfall VoiceType schneller arbeitete als ein Pentium mit 90 MHz. Die neue Version muß nun auf diese Unterstützung verzichten, woraus eine wesentlich höhere Prozessorbelastung resultiert. Eine Erweiterung Ihres PC in Form eines schnelleren Prozessors macht für VoiceType jedoch nur dann Sinn, wenn Sie schon alle anderen Maßnahmen zur Geschwindigkeitssteigerung ausgenutzt haben. Erfüllt Ihr Rechner die minimale Anforderung von einem Pentium mit 100 MHz, können Sie größere Schritte in Sachen Geschwindigkeit erreichen, wenn Sie wie oben beschrieben den Hauptspeicher erweitern oder die Tips für Festplatten beachten. Diese Tuningmaßnahmen wirken sich auch auf die meisten anderen Programme unter OS/2 aus. Mein eigener PC arbeitet mit einem Pentium 100 zu meiner größten Zufriedenheit, allerdings habe ich einen riesigen Arbeitsspeicher und eine extrem schnelle Festplatte.

Festplatte

Die Festplatte hat eine wichtige Bedeutung bei der Arbeit mit VoiceType. Mit ihr steht und fällt die Geschwindigkeit der Spracherkennung und leider wurden schon viele inkorrekte Aussagen über OS/2, VoiceType und die Festplatte gemacht, teilweise sogar in Fachzeitschriften. Deshalb möchte ich nun etwas weiter ausholen, um die Bedeutung der Festplatte zu verdeutlichen.

VoiceType arbeitet mit akustischen Daten. Sobald Sie das Mikrofon zum Diktat anschalten, fallen diese Daten an.

Wie groß ist nun dieses Datenvolumen? Dazu kann man folgende Berechnung anstellen: Die Soundkarte digitalisiert Ihre Sprache als ein Monosignal mit 11,025 kHz und 16 Bit, das heißt pro Sekunde fallen 11025 mal 16 Bit an Daten an. Das sind in der Minute 60*11025*2 Byte = 1,26 Mbyte. Diese Daten fallen beim Diktat an und sind nur abhöngig von der Zeit. Es ist also egal, wieviele Wörter Sie in der Minute diktieren, nur die zeitliche Länge des Diktats ist entscheidend.

Was soll mit diesen Daten geschehen? Wenn Sie nach einem Diktat Korrekturen durchführen, möchten Sie gerne die aufgenommene Akustik wiedergeben. Die akustischen Daten sollen also gespeichert und später wiedergegeben werden. Es gibt nun theoretisch zwei Möglichkeiten, diese Daten zu speichern. Entweder im Arbeitsspeicher oder auf der Festplatte. In der Praxis jedoch wäre jeder Arbeitsspeicher selbst bei nicht allzu langen Diktaten hoffnungslos überfordert. Deshalb wird die akustische Information grundsätzlich auf die Festplatte geschrieben. Das heißt also, daß bei der Arbeit mit VoiceType die Festplatte ständig in Aktion ist. Es han-

delt sich dabei jedoch nicht um Swapping oder Paging (es werden keine Bereiche des Arbeitsspeichers auf die Festplatte geschrieben), sondern um einen simplen Speichervorgang.

Welchen Einfluß hat also die Festplatte auf die Performance von VoiceType? Da VoiceType grundsätzlich immer auf die Festplatte zugreift (beim Diktieren werden Daten geschrieben, beim Korrigieren werden sie gelesen), wirkt sich eine Festplatte sehr stark auf die Geschwindigkeit von VoiceType aus. Je schneller eine Festplatte physisch ist, desto besser verhält sich die Antwortzeit von VoiceType. Aber auch die Art der Formatierung kann sich positiv auswirken. Für derartige Anforderungen, wie VoiceType sie an ein Dateisystem stellt, ist das HPFS (High Performance File System) von OS/2 Warp geradezu prädestiniert. Dies ist auch die einfachste und kostengünstigste Art, VoiceType schneller zu machen: Verwenden Sie das HPFS. Die etwa 500 Kbyte an Arbeitsspeicher, diefür das Einrichten von HPFS benötigt werden, können Sie als VoiceType Benutzer locker »investieren«, Sie haben doch ohnehin bereits mindestens 16 MB RAM. Der Effekt ist, daß sämtliche Ein- und Ausgabe-Operationen schneller vonstatten gehen und VoiceType wesentlich besser reagiert.

Anwendungsprogramme

Tuning von VoiceType kann auch über die Anwendungsprogramme geschehen, die Sie verwenden. Wenn Sie zum Beispiel eine Textverarbeitung verwenden, die auf Ihrem PC schon ohne VoiceType nur nach minutenlangem Warten gestartet werden kann und auf jeden Tastendruck nur mit einiger Verzögerung reagiert, so wird die Antwortzeit katastrophal, wenn Sie zusätzlich noch VoiceType starten.

Dies wird vor allem in Systemen vorkommen, die über relativ wenig Arbeitsspeicher verfügen. Wenn Sie jedoch diesen nicht erweitern wollen oder können, so bleibt Ihnen noch der Ausweg, etwas an Ihrer Software zu verändern.

Sie könnten zum Beispiel grundsätzlich das Diktierfenster verwenden, um Ihre Texte einzugeben, diesen Text dann abspeichern und nach dem Beenden von VoiceType Ihre Zielanwendung starten und den Text importieren. Dies ist zwar nicht die eleganteste Lösung, aber auf jeden Fall diejenige, die den geringsten Speicher benötigt.

Eine andere Möglichkeit wäre, sie wechseln Ihre Zielanwendung. Verwenden Sie ein Programm, das nicht mit unnötigen Funktionen überladen ist, sondern Ihren Bedürfnissen und Anforderungen gerecht wird und dennoch relativ kleinen Speicherbedarf hat. Anwendern, deren PC nur die minimale Größe an Arbeitsspeicher hat, empfehle ich die Textverarbeitung »Papyrus«. Dieses Programm hat einen großen Funktionsumfang und benötigt dennoch nur etwa 1 MB an Arbeitsspeicher. Papyrus wird schneller gestartet als der in OS/2 integrierte erwiterte Editor EPM. Wenn Sie nun durch ein kleineres Programm kostbaren RAM sparen, können Sie gleichzeitig VoiceType starten und mit dem *schnellen Diktat* Ihre Texte erfassen und danach direkt in Ihre Zielanwendung übergeben.

Die letzte und vielleicht eleganteste Lösung wäre, ein Programm zu verwenden, das direktes Diktieren von Text mit VoiceType zuläßt. Bisher ist mir jedoch noch kein Programm bekannt, das die Funktionalität von VoiceType direkt unterstützt. Es bleibt also abzuwarten, wann die ersten Programme von den VoiceType-APIs Gebrauch machen und so direktes Diktieren in die Anwendung möglich wird.

6.17 VoiceType-Upgrade

Die in IBM OS/2 Warp 4 integrierte Version des Diktiersystems VoiceType entbehrt einer Funktion, die vor allem für professionelle Nutzer interessant ist. Sie können nach dem Beenden eines Diktats dieses nicht als »Sitzung« abspeichern, um später die Korrektur durchzuführen oder diese einer anderen Person zu überlassen. Wenn Sie das Upgrade für VoiceType erwerben, können Sie nach der Installation auf diese Funktion zugreifen, so wie es bereits beschrieben wurde.

Entgegen den zunächst bei IBM angestellten Überlegungen können Fachvokabulare für spezielle Anwendungsbereiche in der mitgelieferten Version nun doch installiert werden.

Wenn Sie ein Fachvokabular benötigen, können Sie dies bei Händlern für VoiceType erwerben.

Für die deutsche Version sind derzeit folgende Vokabulare für Spezialisten verfügbar:

– Radiologie

– Gynäkologie

– Orthopädie

– Unfallchirurgie

– HNO

– Recht und Wirtschaft

– Technische Gutachten

Für größere Gruppen von Anwendern ist es möglich, einen eigenen Fachwortschatz erstellen zu lassen. Dazu sind etwa 500 Mbyte ASCII-Text des gewünschten Fachgebietes notwendig. Großkunden oder größere Firmen, die Texte in einem speziellen Fachgebiet erstellen, können mit einem solchen eigenen Wortschatz sehr schnell höchste Erkennungsraten erreichen. Wenn das Erstellen eines Fachwortschatzes für Sie in Frage kommt, wenden Sie sich direkt an IBM.

7 Grundlagen der OS/2-Programmierung

7.1 ReXX

von Anthony Rudd

Jedes Betriebssystem bietet dem Anwender die Möglichkeit, Befehle abzusetzen, das heißt, der Anwender kann Informationen vom Betriebssystem holen oder an das Betriebssystem liefern oder veranlassen, daß das Betriebssystem einen Dienst ausführt. Die Form dieser Interaktion hängt selbstverständlich vom Betriebssystem ab, beispielsweise, ob das Betriebssystem interaktiv arbeitet. Im Lauf der Zeit ist der Bedarf entstanden, eine zusammenhängende Reihe von Kommandos auszuführen, teilweise verbunden mit einfachen logischen Abfragen. In PC-Betriebssystemen sind solche Kommandofolgen als Batchprozeduren oder Skripts bekannt. Auch OS/2 hat eine einfache Batchsprache, die entsprechenden »Programme« werden mit der Erweiterung ».cmd« gekennzeichnet. Solche Reihen von Befehlen können als eine Prozedur betrachtet werden, daher der Begriff »prozedurale Sprache«.

Diese Präambel wurde notwendig, um die Entstehung von ReXX besser verstehen zu können. ReXX steht für das englischen Akronym *Re*structured e*X*tended e*X*ecutor, auf Deutsch: »umstrukturierter erweiterteter Ausführungsmonitor«. Obwohl das ReXX-Akronym nicht viel über seine Fähigkeit oder Einsatzmöglichkeiten sagt, erklärt es, warum ReXX oft in Zusammenhang mit Königen steht. ReXX wurde ursprünglich REX genannt, und das lateinische Wort »rex« bedeutet König. ReXX ist eine sehr mächtige Sprache, deren Fähigkeit oft unterschätzt wird. Dieses Kapitel sollte dazu dienen, ReXX und deren Verwendung besser zu verstehen.

Ein kurzer Rückblick: ReXX wurde Anfang der 80er Jahren von M.F. Cowlishaw, einem IBM-Mitarbeiter, als Erweiterung der CMS-Kommandosprache für das IBM-VM-Betriebsystem entwickelt. Sein Ziel war, eine natürliche Sprache zu entwickeln. Im Gegensatz zu vielen Programmiersprachen, war die einfache Handhabung von ReXX wichtiger als die einfache Implementierung.

7.1.1 Einführung

Ein Kapitel reicht nicht aus, um eine Programmiersprache vollständig zu erklären, besonders für eine Sprache wie ReXX. Ich kann lediglich die wichtigsten Merkmale beschreiben und einige Einsatzmöglichkeiten erwähnen, so daß Sie als Leser ein Gefühl bekommen, was die Sprache ist und wie sie eingesetzt werden kann.

Das erste Programm

Um die ReXX-Konzepte plastisch darzustellen, werden wir mit einem einfachen Programm beginnen, und das Programm sukzessive ausbauen.

Wie fast üblich, fangen wird mit dem allgegenwärtigen »Hello World«-Programm an. Dieses einfache ReXX-Programm begrüßt den Anwender:

```
SAY HELLO WORLD
```

Viel einfacher könnte es nicht sein. Die Say-Anweisung gibt alle Wörter aus, die der Anweisung folgen. Eigentlich sind »HELLO« und »WORLD« Variablen. Variablen in ReXX müssen nicht (und können auch nicht) deklariert werden und haben standardmäßig ihren eigenen Namen (in Großbuchstaben) als Inhalt. Insofern entspricht das Programm nicht ganz der Anforderung nach einem saubereren Programmierstil. Richtiger wäre folgende Zuweisung:

```
SAY "Hello World"
```

Hier wurde ein Literal verwendet. Um das Programm zu verallgemeinern, könnte man den Namen des Bedieners von der Tastatur holen, also durch den Anwender eingeben lassen. Der befehl »Pull« (= heranziehen) holt eine Datenzeile vom Terminal. Dazu muß nur die folgende Zeile in das Programm eingefügt werden:

```
PULL bediener
```

Das gesamte Programm mit Ein- und Ausgabe sieht dann so aus:

```
/* ReXX */
SAY "Bitte Namen eingeben:"
PULL bediener
SAY "Ihr Name ist" bediener
```

Der ReXX-Kommentar als erste Zeile des Quelltexts ist notwendig, damit der OS/2-Kommandointerpreter »cmd.exe« eine ReXX-Prozedur von einer normalen Batch-Prozedur unterscheiden kann. Aber auch hier in diesem Listing befindet sich ein kleinerer Fehler. Der Befehl »Pull« ist eigentlich eine Abkürzung für »PARSE-PULL-UPPER«. Das führt dazu daß implizit eine Umsetzung auf Großbuchstaben (das UPPER-Schlüsselwort) erfolgt. Der explizite »Parse«-Befehl mit »Pull« als zusätzlichem Schlüsselwort löst dieses Problem. Die Anweisung sollte dann lauten:

```
PARSE PULL bediener
```

Die Anweisung »Parse« (= analysieren) ist wahrscheinlich die mächtigste ReXX-Anweisung. Bisher ist nur die einfachste Variante verwendet worden – eine einzige Variable (hier »Bedie-

ner«) wurde extrahiert. Eine andere Variante wäre, den Nach- und Vornamen einzeln zu holen. Hier müssen zwei Variablen spezifiziert werden (ein Leerzeichen wird implizit als Trennzeichen verwendet).

```
PARSE PULL nachname vorname
```

Zum Beispiel könnte »Meier Hans« hier als Eingabe spezifiziert werden. Es ist auch möglich, ein explizites Trennzeichen zu verwenden.

```
PARSE PULL nachname"," vorname
```

In diesem Fall muß die Eingabe im Form von »Meier,Hans« gemacht werden.

Wir werden dieses Programm mit einer Schleife für die Eingabe von Namen und Adresse erweitern. Um Schleifen- und Vergleichskonstrukte zu zeigen, können maximal fünf Eingabezeilen spezifiziert werden. Eine leere Eingabe beendet die Schleife.

```
/* ReXX */
SAY "Namen und Adresse eingeben"
DO i = 1 TO 5
  SAY "Zeile" i "eingeben"
  PARSE PULL zeile.i
  IF zeile.i = '' THEN LEAVE
  imax = i
END

SAY imax 'Zeilen wurden eingegeben'
```

Hier wurde eine neue Art von Variable verwendet: eine Compound-Variable (sogenannte Stem-Variablen). Eine Compound-Variable besteht aus zwei oder mehr Bezeichnern (»Qualifiern«). Diese Bezeichner werden durch einen oder mehrere Punkte getrennt. Der erste Bezeichner wird als »Stem« bezeichnet. Die weiteren Bezeichner entsprechen in etwa einem Index einer Struktur, obwohl in ReXX solche Bezeichner nicht nur numerisch sein müssen. Die imax-Variable enthält die Anzahl der gespeicherten Zeilen.

Diese Erweiterung verwendet vier neue Anweisungen:

– DO ... END

– IF

– LEAVE

– = (Zuweisung)

Da diese Anweisungen die übliche programmtechnische Bedeutung haben, werden sie hier nicht weiter erläutert.

Größere Programme können vereinfacht werden, indem sie in kleinere Einheiten (sprich: Routinen) unterteilt werden.

Wir werden unser Programm nochmals erweitern, um die eingegebenen Namen-Adreß-Zeilen abzuspeichern. Obwohl Variablen Werte für die Lebenszeit des ReXX-Programmes speichern

können, können solche Variablen nur in externen Programmen direkt verwendet werden. Andere Routinen können Kopien von Variablen verwenden, entweder als Laufparameter (beziehungsweise für als Speicher des Ergebnisses der Routine) oder unter Verwendung der ReXX-Queue. Das erweiterte Programm verwendet alle drei Arten.

```
CALL Speichern imax
```

Diese Call-Anweisung ruft das Unterprogramm »Speichern« mit »imax« als Parameter auf, um die eingegebenen Zeilen in die ReXX-Queue zu stellen.

```
Speichern:
PROCEDURE EXPOSE zeile.
  PARSE ARG n /* Anzahl Zeilen */
  DO i = 1 TO n
    QUEUE zeile.i
  END
RETURN n
```

Das Schlüsselwort »Expose« ist notwendig, um der Prozedur »Speichern« Zugriff auf die Variable »Zeile.« zu geben. Die Anweisung »Parse« mit dem Schlüsselwort »Arg« stellt die spezifizierte Parameterliste in die angegebene Variable (»N«). Die folgende Schleife stellt die Variable »zeile.« in die ReXX-Queue. Die Queue-Anweisung stellt dabei das Element immer an das Ende der Queue (die verwandte Push-Anweisung stellt das Element an den Anfang der Queue). Zum Schluß beendet die Return-Anweisung das Unterprogramm und gibt die Steuerung mit der Anzahl der gespeicherten Elemente als Ergebnis zurück. Das aufrufende Programm kann das Unterprogrammergebnis aus der Variablen »Result« holen.

```
SAY RESULT "Zeilen gespeichert"
EXIT /* beenden */
```

Das Gesamtprogramm sieht dann wie folgt aus:

```
/* ReXX */
SAY "Namen und Adresse eingeben"
DO i = 1 TO 5
  SAY "Zeile" i "eingeben"
  PARSE PULL zeile.i
  IF zeile.i = '' THEN LEAVE
  imax = i
END
CALL Speichern imax
SAY RESULT "Zeilen gespeichert"
EXIT                                  /* Beenden       */

Speichern: PROCEDURE EXPOSE zeile.
  PARSE ARG n                         /* Anzahl Zeilen */
  DO i = 1 TO n
    QUEUE zeile.i
  END
RETURN
```

Um die Fähigkeit der Programmiersprache ReXX zeigen zu können, müssen wir ein etwas aufwendigeres Beispiel entwickeln. Wir bleiben bei der Terminal-Ein- und -Ausgabe, obwohl auch Dateien in ReXX verarbeitet werden können.

Das zweite Programm

Ein ReXX-Programm wird interpretiert statt kompiliert. Das hat Vor- und Nachteile. Eine interpretierte Sprache wird bei jeder Ausführung neu analysiert, da kein ausführbarer Objektcode zur Verfügung steht. Dies bedeutet normalerweise, daß die Ausführungsgeschwindigkeit geringer als bei einem kompilierten Programm. Dafür entfallen aber aufwendige Kompilierungs- und Bindeläufe. ReXX versucht einige der Nachteile eines Interpreters damit wettzumachen, indem ein Zwischencode erzeugt und im Speicher gehalten wird. Dieser Zwischencode erspart einigen Analyseaufwand, so daß das Programm schneller abgearbeitet werden. ReXX als interpretierte Sprache bietet den Vorteil, daß sich das Programm zur Laufzeit modifizieren kann. Diese Fähigkeit wird in folgendem zweiten Programmlisting verwendet.

Das Programm hat die Aufgabe, einen einfachen (Taschen-)Rechner darzustellen. Eine Eingabezeile enthält einen arithmetischen Ausdruck. Dieser Ausdruck wird ausgewertet und das Ergebnis ausgegeben.

Wie immer, soll der Anwender informiert werden, was er machen muß. Wie bereits gezeigt, verwenden wir die Say-Anweisung »SAY "arithmetischen Ausdruck eingeben"«. Die Parse/Pull-Anweisung wartet auf die Eingabe.

```
PARSE PULL ausdruck
```

ReXX bietet mit der Anweisung »Interpret« die Möglichkeit, zur Laufzeit eine Anweisung auszuführen. In unserem Fall müssen wir lediglich eine Anweisung mit dem spezifizierten Ausdruck erstellen und ausführen, um das Ergebnis des Ausdrucks zu bekommen.

```
stmt = "x =" ausdruck
INTERPRET stmt
SAY "Ergebnis:" x
```

Die Laufzeit-Anweisung wird in die Variable »stmt« gestellt, die anschließend bearbeitet (Anweisung »Interpret«) wird.

Im Prinzip ist das Programm hiermit lauffähig, vorausgesetzt, der Anwender macht fehlerfreie Angaben. Wird nämlich ein syntaktisch falscher Ausdruck eingegeben, bricht ReXX mit einem Laufzeitfehler ab.

Um eine kontrollierte Fehlermeldung auszugeben, kann eine Fehlerbehandlung in das Programm eingebaut werden, die Falscheingaben abfängt. Die Anweisung »Signal« gibt hier die Steuerung an *Fehler*, falls ein Syntax-Fehler vorliegt. Das Unterprogramm »Fehler« gibt eine Meldung mit der Zeilennummer (Variable »sigl«) aus, in der der Fehler aufgetreten ist. Dazu das Gesamtprogramm:

```
/* ReXX */
SAY "arithmetischen Ausdruck eingeben"
PARSE PULL ausdruck
```

```
stmt = "x =" ausdruck
SIGNAL ON SYNTAX NAME Fehler
INTERPRET stmt
SAY "Ergebnis:" x
EXIT
Fehler:
  SAY "Syntax Fehler in Zeile" SIGL
```

Wir haben jetzt einen Überblick über ein paar einfache ReXX-Programme und einen Eindruck von der Sprache gewonnen. Allerdings eine Bemerkung im voraus: Diese Programme sind nicht besonders typisch, weder für ReXX und ganz besonders nicht für OS/2-ReXX.

Außerdem wurden nur einige ReXX-Befehle beschrieben.

Ähnlich wie C bietet ReXX eine umfangreiche Standardfunktionsbibliothek. Diese Standardfunktionsbibliothek enthält Funktionen für Zeichenfolgen, Dateiverarbeitung und Allgemeines. OS/2-ReXX bietet zusätzlich dann auch noch eine Standardbibliothek von Hilfsfunktionen, die speziell auf die OS/2-Umgebung abgestimmt sind (um beispielsweise ein Workplace-Shell-Objekt zu erstellen). ReXX bietet auch eine nahtlose Schnittstelle zur aktuellen Laufzeitumgebung (wie Betriebssystemdienste oder produktspezifische Dienste). Diese Funktionalität wird später in diesem Kapitel beschrieben, aber zuerst sehen wir uns die ReXX-Sprachelemente etwas genauer an.

7.1.2 Die Sprachelemente von ReXX

Ein ReXX-Programm besteht aus einer oder mehreren Anweisungen. Außer in Literalen unterscheidet ReXX nicht zwischen Groß- und Kleinschreibung. So sind die Bezeichner »Alpha«, »alpha« und »ALPHA« identisch; »"Alpha"« und »"ALPHA"« stellen aber zwei unterschiedliche Zeichenfolgen dar. Ich persönlich schreibe Variablennamen normalerweise klein, um die Lesbarkeit der Quelltexte zu erhöhen.

Klauseln

Jede ReXX-Anweisung besteht aus einer oder mehreren (mit Strichpunkten getrennten) Klauseln. Eine Klausel besteht aus Tokens. Leerzeichen zwischen Tokens werden ignoriert. Ein Kommentar hat die gleiche syntaktische Bedeutung wie ein Leerzeichen. Die Bedeutung beziehungsweise Form der Klausel hängt von der Operation ab. Viele Operationen erlauben Ausdrücke als Klauseln.

Kommentar

Ein Kommentar kann beliebige Zeichen (außer »/*« und »*/«) innerhalb der Zeichenpaare »/*« (Kommentaranfang) und »*/« (Kommentarende) enthalten. ReXX-Kommentare können verschachtelt werden. Ein noch offener Kommentar am Ende der Zeile wird implizit fortgesetzt. Ein explizites Fortsetzungszeichen bei mehrzeiligen Kommentaren ist also nicht notwendig:

```
/* ein verschachtelter /* Kommentar */ */
/* ein fortgesetzter
   Kommentar */
```

Tokens

Jede Anweisung besteht aus einem oder mehreren (mit einem Semikolon getrennten) Tokens. Ein Token ist das kleinste syntaktische Element. Tokens sind:

– Literale

– Symbole

– Operatoren

– Sonderzeichen

Literale

ReXX kennt drei Arten von Literalen (Zeichenfolgen):

– normale Zeichenfolgen

– hexadezimale Zeichenfolgen

– binäre Zeichenfolgen

Ein Literal in ReXX wird in Hochkommas (') oder Anführungszeichen ("") eingeschlossen. Obwohl das Anfangsbegrenzungssymbol auch als Endbegrenzungssymbol verwendet werden muß, kann das andere Begrenzungssymbol ohne Einschränkungen innerhalb eines Literals verwendet werden. »X« oder »B« als Suffix kennzeichnen ein hexadezimales beziehungsweise binäres Literal. Offene Literale werden implizit fortgesetzt.

Wird das Begrenzungssymbol innerhalb der Zeichenfolge gebraucht, muß es verdoppelt werden, zum Beispiel:

"alpha"

"alpha' (falsch – ungültiges Endbegrenzungssymbol)

"al'pha"

'IBM''s' (entspricht IBM's, obwohl "IBM's" einfacher wäre)

"1234ab"X (hexadezimale Zeichenfolge)

"01001"B (binäre Zeichenfolge).

Hexadezimale Zeichenfolgen

Eine hexadezimale Zeichenfolge ist ein Literal mit einem unmittelbar angehängten »X« (oder »x«). Eine hexadezimale Zeichenfolge kann nur die Ziffern 0 bis 9 sowie die Buchstaben »a« bis »f und »A« bis »F« enthalten. Um die Lesbarkeit zu verbessern, dürfen Leerzeichen zwi-

schen Zeichenpaaren erscheinen. Eine Null wird vor den Anfang einer hexadezimalen Zeichenfolge angehängt, die eine ungerade Anzahl von Ziffern besitzt. Hierzu einige Beispiele:

"6162"X entspricht "ab"

"12 34 ab"X

"123"X entspricht "0123"X

Falsch wäre aber ein »"12g"X« Die Zeichenkette enthält ein ungültiges hexadezimale Zeichen.

Binäre Zeichenfolgen

Eine binäre Zeichenfolge ist ein Literal mit einem unmittelbar angehängten »B« (oder »b«) . Eine binäre Zeichenfolge darf nur die Ziffern 0 und 1 enthalten. Um die Lesbarkeit zu verbessern, dürfen zwischen Gruppen von vier Ziffern Leerzeichen erscheinen. Führende Nullen werden am Anfang einer binären Zeichenfolge angehängt, falls die Anzahl von Ziffern nicht ein Vielfaches von 8 ist. Beispiele hierzu wären:

"0011 0001 0011 0010"B entspricht "12"

"0001 0011 0010"B entspricht "0132"X

Eine Angabe von »"1200"B« wäre falsch, da die Zahl 2 in einer binären Zeichenfolge nicht erscheinen darf.

Symbole

Symbole sind Buchstaben aus dem englischen Alphabet (a bis z, A bis Z), numerische Ziffern (0 bis 9) und die Sonderzeichen ».«, »!«, »?« und »_«. Umlaute sind in Symbolen nicht möglich.

In Symbolen werden Kleinbuchstaben automatisch in Großbuchstaben umgesetzt. Die Darstellung eines exponentiellen Wertes (eine Zahl zur Basis 10) ist die einzige Ausnahme. Die Bedeutung eines Symbols hängt von dessen Verwendung ab. Beispiel für Symbole wären »alpha«, »umsatz.jahr«, »123.4« oder 2e+3. Letzteres ist ein exponentieller Wert, der der Zahl 2000 entspricht.

Operatoren

Operatoren werden verwendet, um Ausdrücke zu bilden. ReXX kennt folgende Arten von Operatoren:

– arithmetische Operatoren

– logische Operatoren

– Vergleichsoperatoren

– Verkettungsoperatoren

– Präfixoperatoren

Ein Operator erstellt den entsprechenden Ausdruck. Zum Beispiel bildet ein arithmetischer Operator einen arithmetischen Ausdruck, obwohl die Ausdrucksart nicht eindeutig sein muß. So kann ein arithmetischer Ausdruck mit dem Wert 0 oder 1 auch als logischer Ausdruck verwendet werden.

Arithmetische Operatoren

ReXX kennt alle wichtigen arithmetischen Operatoren:

Operator	Bedeutung
**	Exponentiation (nur ganzzahlig)
*	Multiplikation
/	Division
//	ganzzahlige Division
%	Rest einer Division
+	Addition
-	Subtraktion

Ein arithmetischer Ausdruck hat das Format:

```
Operand1 Operator Operand2
```

Operand1 und *Operand2* sind numerische Werte; *Operator* ist ein arithmetischer Operator. Da das Ergebnis eines arithmetischen Ausdrucks selbst ein numerischer Wert ist, kann ein arithmetischer Ausdruck in einem weiteren arithmetischen Ausdruck verwendet werden. ReXX verwendet die übliche Priorität für arithmetische Operatoren, Exponentialrechnung hat also Vorrang über multiplikative (»*«, »/«, »//«, »%«) Operatoren, die wiederum Vorrang über additiven Operatoren (»+«, »-«) haben. Klammern werden verwendet, um explizite Prioritäten zu setzen. Hierzu einige Beispiel:

```
7 + 2       -> 9
7 – 2       -> 5
7 * 2       -> 14
7 ** 2      -> 49
7 / 2       -> 3.5
7 // 2      -> 3
7 % 2       -> 1
7 + 2 * 3   -> 13 = (2 * 3) + 7
(7 + 2) * 3 -> 27
```

Wichtig ist, zu wissen, daß die Präfixoperatoren (»+« und »-«) Vorrang haben über alle arithmetischen Operatoren. Dazu ein Beispiel:

```
3 ** -2     -> 0.111111
```

Logische Operatoren (Boolesche Operatoren)

ReXX kennt alle notwendigen logischen Operatoren:

Operator	**Bedeutung**
\	Nicht (Negation)
&	UND
		ODER
&&	Antivalenz

Ein logischer Ausdruck hat das Format:

```
Operand1 BinOperator Operand2
```

beziehungsweise

```
Unioperator Operand1
```

Operand1 und *Operand2* sind binäre Werte, das heißt 0 oder 1. *BinOperator* ist ein binärer Operator, also ein Operator, der zwei Operanden besitzt (»|«, »&«, »&&«). *UniOperator* ist ein unärer Operator, also ein Operator, der einen Operand besitzt (»\«).

Das Ergebnis eines logischen Ausdrucks ist ein binärer Wert. Wie bei arithmetischen Ausdrücken kann ein logischer Ausdruck selbst wieder in einem logischen Ausdruck verwendet werden. Negation hat Vorrang über »&«, der Vorrang über »&&« und »|« hat. Hierzu einige Beispiele:

```
1 | 0   -> 1
0 | 1   -> 1
1 | 1   -> 1
0 | 0   -> 0
1 & 0   -> 0
0 & 1   -> 0
1 & 1   -> 1
0 & 0   -> 0
1 && 0  -> 0
0 && 1  -> 1
\ 1     -> 0
\ 0     -> 1
\ 1 | 0 -> 0
```

Boolesche Ausdrücke werden normalerweise verwendet, um Wahrheits-Ausdrücke zu bilden. Hierzu wieder ein Beispiel:

```
IF (a%2 & a>0) THEN SAY "A ist positiv und gerade"
IF (a%2 & a>0) | (b%3 & b<0)
  THEN SAY "A positiv und gerade oder B negativ und ungerade"
```

Vergleichsoperatoren

ReXX kennt eine große Anzahl von Vergleichsoperatoren:

Operator	**Bedeutung**
=	gleich
==	genau gleich
>	größer als
>>	genau größer als
>=	größer als oder gleich
>>=	genau größer als oder gleich
<	kleiner als
<<	genau kleiner als
<=	kleiner als oder gleich
<<=	genau kleiner als oder gleich
¬=	ungleich (auch \=, <> und ><)

Als Ergänzung zu den üblichen Vergleichsoperatoren kennt ReXX zusätzlich noch den »Genau«-Qualifier. Der »Genau«-Qualifier berücksichtigt Leerzeichen, die am Anfang oder am Ende eines Operands stehen. Ein Vergleichsausdruck liefert den (binären) Wahrheitswert als Ergebnis (0 = unwahr, 1 = wahr). Einige Beispiele:

```
"a " = "a"          -> 1 (wahr)
"a " == "a"         -> 0 (unwahr)
"a " >= "a"         -> 1 (wahr)
"a " >>= "a"        -> 0 (unwahr)
```

Verkettungsoperatoren

ReXX verwendet »||« für die explizite Verkettung. Die zwei Operanden werden direkt miteinander verkettet. Der Leerzeichen-Operand führt eine implizite Verkettung durch; die zwei Operanden werden mit einem einzigen Leerzeichen dazwischen miteinander verkettet, zum Beispiel:

```
"ab" || "cd"    -> "abcd"
"ab" ,
|| "cd"         -> "abcd"
"ab"     "cd"   -> "ab cd"
"ab" ,
    "cd"        -> "ab cd"
```

aber

```
"ab""cd"         -> "ab"cd"       /* Literal */
```

Sonderzeichen

ReXX kennt die die folgenden vier Sonderzeichen:

Operator	Bedeutung
;	Ende einer Anweisung (nur notwendig wenn eine physikale Zeile mehrere Anweisungen enthält)
,	explizite Fortsetzung, Trennung zwischen den Argumenten
einer	Routine
:	Ende einer Marke
( )	Kennung für einen Funktionsaufruf, explizite Priorisierung eines Ausdrucks

Hierzu einige Beispiele:

```
a = 1; b = 2            /* 2 Anweisungen in einer Zeile    */
a = 1 ,                 /* Komma als Fortsetzung           */
+ 2
a = LEFT(b,4)           /* Komma als Operandentrennung     */
PARSE ARG p1,p2         /* Komma als Operandentrennung     */
Namenslaenge: PROCEDURE  /* Doppelpunkt als Markenende      */
a = LENGTH(b)           /* ( ) als Funktionsaufrufkennung  */
a = (3+4)*5             /* ( ) als explizite Priorisierung */
```

Variablen

ReXX-Variablen werden nicht deklariert und haben nur einen einzigen Typ, nämlich »String« (Zeichenkette). Ob eine Variable numerischen, alphanumerischen oder anderen Typs ist, hängt vom aktuellen Inhalt der Variablen ab. Die aktuelle Einstellung »NUMERIC DIGITS« spezifiziert die Genauigkeit eines numerischen Werts (der Standardwert ist 9). Ein Variablenname hat eine maximale Länge von 250 Zeichen.

ReXX kennt zwei Arten von Variablen:

– Einfache Variablen

– Compound-Variablen

Eine »einfache Variable« ist ein Name, der von den folgenden Zeichen gebildet wird:

– a – z Kleinbuchstaben

– A – Z Großbuchstaben

– 0 – 9 Ziffern

– !, ?, _ Sonderzeichen

Eine Ziffer darf nicht als erstes Zeichen eines Variablennamens verwendet werden.

Obwohl eine ReXX-Variable standardmäßig ihren Namen (in Großbuchstaben) als Inhalt hat, ist diese »Erleichterung« die häufigste Fehlerquelle in ReXX-Programmen, da der Variablen später im Programm ein Wert zugewiesen wird. Es ist also besser, ein Literal zu verwenden.

ReXX besizt drei reservierte Variablen-Namen:

RC

Return-Code. Das Ausführungsergebnis eines Befehls, nicht zu Verwechseln mit dem Ergebnis des Befehls.

iRESULT

Das Ergebnis einer Routine. Der Wert, der mit der Anweisung »RETURN« zurückgegeben wird (= Funktionswert).

SIGL

»sigl« steht für »Signal«. Diese Variable enthält die Zeilennummer der letzten Anweisung im Quellprogramm, die den Sprung direkt (mit einer Signal- oder Call-Anweisung oder einem Funktionsaufruf) oder indirekt (Ausnahmebedingung) abgesetzt hat.

Eine Compound-Variable besteht aus zwei oder mehrere Bezeichnern. Diese werden durch einen oder mehrere Punkte getrennt. Der erste Bezeichner (»Stem«) ist fest, d.h., er behält seinem Namen, auch wenn einer einfachen Variablen mit dem gleichen Namen ein Wert zugewiesen wurde. Die weiteren Bezeichner werden wie normale Variablen behandelt.

Eine Compound-Variable entspricht (in etwa) einem Record in anderen Programmiersprachen. Außer der Einschränkung in der maximalen Namenslänge ist die Anzahl der Dimensionen (Folge-Qualifier) begrenzt. In Gegensatz zu anderen Programmiersprachen müssen die Folge-Qualifier nicht unbedingt numerisch sein (numerische Folge-Qualifier müssen auch nicht fortlaufend sein). ReXX-Programme machen normalerweise häufig Gebrauch von dieser Fähigkeit. Eine einfache Art von assoziativem Speicher kann mit dieser Fähigkeit implementiert werden (siehe Beispiel 2).

Obwohl ReXX-Variablen nicht deklariert werden können, können sie initialisiert werden. Die Zuweisung eines Wertes initialisiert die Variable. Compound-Variablen können ebenfalls initialisiert werden, auch ihre Instanzen, die noch nicht existieren.

Beispiel 1:

```
monat = 2
jahr = 1996
temp.monat.jahr = 10
```

und

```
temp.2.1996 = 10
```

sind äquivalent.

Beispiel 2 (assoziativer Speicher):

```
Wortanzahl. = 0                    /* initialisieren      */
zaehler = 0                        /* Anzahl Einzelwörter */
DO ...
                       /* »wort« enthält die Einzelwörter */
  IF (Wortanzahl.wort = 0) THEN DO          /* neues Wort */
  zaehler = zaehler+1 /* Anzahl der Wörter inkrementieren */
```

```
  Woerterbuch.zaehler = wort            /* Wort speichern */
  END
  Wortanzahl.wort = Wortanzahl.wort+1   /* Worthäufigkeit */
END
```

Dieses Beispiel speichert die Einzelwörter eines Dokuments in der Compound-Variablen *Woerterbuch.* (um die Logik zu vereinfachen, wird die Analyse des Dokuments und die Zuordnung der Einzelwörter weggelassen). Die Häufigkeit der Einzelwörter wird in der Compound-Variablen *Wortzahl.* gespeichert.

Marken (Labels)

Eine Marke ist ein Sprungziel und identifiziert eine Routine oder Anweisung. Eine Marke wird durch einen angehängten Doppelpunkt gekennzeichnet:

```
CALL Unterprogramm
  /* ... */
Unterprogramm: PROCEDURE
```

Routinen

Eine Routine ist eigentlich eine Gruppe von Anweisungen. Ein ReXX-Programm ist eine (externe) Routine. Routinen können auch interne Routinen besitzen. Eine interne Routine wird durch eine Marke, die dem Routinennamen entspricht, identifiziert.

ReXX kennt zwei Arten von Routinen:

– Unterprogramme

– Funktionen

Unterprogrammen und Funktionen unterscheiden sich dadurch, daß eine Funktion ein Ergebnis zurückliefert. Eine Prozedur (PROCEDURE-Instruktion) ist eine spezielle Art von Routine, in der die Variablen nur innerhalb der Routine bekannt sind (sichtbare Variablen müssen mit *Expose* explizit genannt werden). Unterprogramme können explizit mit der Anweisung »Call« aufgerufen werden. Funktionen können mit der Call-Anweisung oder mit einem Funktionsaufruf aufgerufen werden. Die Anweisung »Return« verläßt eine Routine.

Die Routinen können entweder intern (enthalten im physikalischen Programm) oder extern sein. Externe Routinen können in ReXX oder in einer konventionellen Programmiersprache wie C geschrieben werden:

```
n = 0
CALL rtn1
SAY n /* 1 */
b = rtn2()
SAY m n /* 3 1 */
..
EXIT

rtn1:
  n = 1
```

```
...
RETURN

rtn2: PROCEDURE EXPOSE m
  m = 3
  n = 2
RETURN 0     /* Funktionsende */
```

Funktionsaufrufe

Obwohl ein Funktionsaufruf kein eigenständiges ReXX-Sprachelement ist, wird er häufig in Ausdrücken verwendet. Der Funktionsaufruf für eine Funktion, die einen Funktionswert zurückgibt, kann wie ein entsprechender Wert verwendet werden, beispielsweise

```
a = "ab cd"
b = LENGTH(a)+2    /* 7 */
```

Die ReXX-Queue

Die ReXX-Queue ist ein interner Puffer. Elemente können am Anfang (»Push«) oder am Ende (»Queue«) der Queue gespeichert werden. Elemente können aber nur vom Anfang des Puffers geholt werden. Die ReXX-Queue ist unabhängig von der OS/2-Queue.

ReXX macht häufig Gebrauch von Queues, um Daten zwischenzuspeichern. Da benannte Variablen nicht zwischen externen Routinen angesprochen werden können, werden Queues dazu verwendet, größere Datenmengen zwischen externen Routinen zu übergeben. Obwohl eine ReXX-Anwendung mehrere Queues besitzen kann, darf immer nur eine Queue aktiv sein. Die Funktion »RxQueue« ist zuständig für die allgemeine Queue-Verwaltung (Queue anlegen, Queue aktivieren, Queue löschen usw.). Der Filter »RxQueue« wird verwendet, um die Bildschirmausgabe in die Queue umzuleiten. Die Standard-Queue hat den Namen »Session«. Dazu das obligarotische Beispiel:

```
qn = RXQUEUE('CREATE','USER')              /* Queue anlegen */
                          /* qn = angelegter Name der Queue */
"DIR C: | RXQUEUE" qn "/LIFO"       /* DIR-Ausgabe umleiten */
CALL  RXQUEUE 'SET',qn               /* aktive Queue setzen */
PULL zeile
PARSE VAR zeile freibytes .
SAY freibytes 'Bytes frei'
CALL RXQUEUE 'DELETE',qn                   /* Queue löschen */
```

7.1.3 Anweisungen

ReXX kennt die folgenden Operationen (sogenannte Schlüsselwort-Instruktionen):

Anweisung	**Bedeutung**
ADDRESS	Umgebung setzen
ARG	Argument holen
CALL	Routine aufrufen
DO...END	Schleife
DROP	Variable freigeben
EXIT	Prozedur beenden
IF...THEN...ELSE	Bedingung prüfen
INTERPRET	Laufzeitanweisung ausführen
ITERATE	Schleife nochmals durchlaufen
LEAVE	Schleife verlassen
NOP	keine Operation
NUMERIC	numerische Formate und Genauigkeit bestimmen
OPTIONS	Umgebungsoptionen
PARSE	Daten analysieren
PROCEDURE	Unterprogrammprozedur
PULL	Erstes Element der Queue holen
PUSH	Element an den Anfang der Queue stellen
QUEUE	Element an das Ende der Queue stellen
RETURN	Routine verlassen
SAY	Anzeigen (Textausgabe)
SELECT...WHEN...OTHERWISE...END	Bedingte Selektion einer Anweisung aus einer Gruppe von Anweisungen
SIGNAL	Ausnahmebedingung aktivieren bzw. deaktivieren oder die Steuerung an eine Routine übergeben
TRACE	Laufzeitinformationen anzeigen (Testhilfe)
=	Zuweisung

Mit Ausnahme der Zuweisungsoperation (»=«) und einem Befehl, stellen die oben aufgelisteten Operationen immer das erste Token (syntaktisches Element) einer Anweisung dar; die Ergebnisvariable ist das erste Token in einer Zuweisungsoperation. Ein Befehl ist eine Anweisung, die keine ReXX-Operation besitzt. Ein Befehl wird an die aktuelle Umgebung übergeben.

Eine Anweisung kann mit einem Semikolon (»;«) beendet werden. Da eine physikalische Programmzeile implizit mit einem Semikolon endet, ist diese Angabe nicht notwendig, es sei denn, eine physikalische Programmzeile enthält mehrere Anweisungen. Dies ist aber auf Grund der Übersichtlichkeit normalerweise nicht zu empfehlen; diese Schreibweise sollte nur für Anweisungen verwendet werden, die sehr eng miteinander verknüpft sind. Ein Beispiel:

```
a = 1; b = 2  /* Das Semikolon schließt die 1. Anweisung ab */
```

Ein Komma muß spezifiziert werden, falls eine Anweisung auf einer Folgezeile fortgesetzt wird und die Fortsetzung nicht implizit ist. Zum Beispiel ist bei

```
IF a = 1
  THEN b = 2
```

ein explizites Fortsetzungszeichen nicht notwendig, weil auf die IF-Klausel von eine THEN-Klausel folgen muß. Im Fall von

```
x = a,
  + 1
```

muß ein explizites Fortsetzungszeichen spezifiziert werden.

ADDRESS – Umgebung setzen

Die Anweisung »Address« setzt die aktuelle Umgebung. Eine Anweisung ohne Schlüsselwort-Instruktion (oder Zuweisung) wird an den aktuellen Umgebungshandler übergeben. Der Standardhandler (OS/2-Befehlsinterpreter) hat den Namen »CMD« für »cmd.exe«.

Viele Fremdprodukte (z.B. das Datenbanksystem DB2/2) stellen einen eigenen Umgebungshandler zur Verfügung. Der Anwender kann außerdem seine eigenen Umgebungshandler schreiben (in einer konventionellen Programmiersprache wie C).

```
ADDRESS "CMD"
"DIR C:"
```

ARG – Argument holen

Diese Anweisung weist die Argumente eines Aufrufs einer Variablen zu. Die Arg-Anweisung ist eine Untermenge der Anweisung »Parse Arg«. Dazu ein Beispiel:

```
CALL Funkt 0 "CMD"
/* ... */
Funkt:
  ARG p1 p2
SAY p1 /* 0 */
SAY p2 /* CMD */
  ...
```

CALL – Routine aufrufen

Die Call-Anweisung ruft eine Routine auf. Dabei können Argumente übergeben werden. Die Variable »Result« enthält das Ergebnis der Routine:

```
CALL Funkt
SAY RESULT /* Ergebnis von »Funkt« */
```

DO...END-Schleife

Die DO...END-Anweisung definiert eine Schleife.

ReXX bietet die folgenden Schleifenarten:

– Einfach (wird nur einmal durchgeführt und wird oft anstelle einer Anweisung verwendet).

– Endlos (Forever-Anweisung)

– Bedingte Schleife (Bedingung »While«- bzw. »Until«)

– Wiederholungsschleife (wahlweise mit To-Bedingung oder For-Anweisung)

Dazu einige Beispiele:

```
DO I = 1 TO 10
  /* ... */
END

DO FOREVER /* Endlosschleife */
  /* ... */
END
```

DROP – Variable freigeben

Die Drop-Anweisung gibt eine Variable frei, damit ist die Variable nicht mehr initializiert; zugeordnete Speicherbereiche sind freigegeben. Das ist insbesondere bei großen Compound-Variablen interessant:

```
DROP varn
```

EXIT – Prozedur beenden

Diese Anweisung beendet eine Prozedur. Ein Wert kann zurückgegeben werden. Eine Prozedur wird standardmäßig am Ende des Programms beendet. Das Beispiel hierzu:

```
CALL Funkt
..
EXIT /* notwendig, da sonst »Funkt« nochmals aufgerufen wird */
Funkt:
..
```

IF...THEN...ELSE – Bedingung prüfen

Die IF...THEN...ELSE-Anweisung prüft eine Bedingung. Die Else-Klausel ist wahlweise. Da die Then- bzw. Else-Klausel nur Anweisungen sind, muß notfalls eine einfache Do-Schleife verwendet werden.

Das Beispiel hierzu:

```
IF a < 0
  THEN DO
    /* Bedingung erfüllt */
  END
  ELSE SAY "Bedingung nicht erfüllt"
```

INTERPRET – Laufzeitanweisung ausführen

Die Interpret-Anweisung führt zur Laufzeit eine Anweisung aus. Sie wird verwendet, um eine dynamisch erstellte Anweisung auszuführen (z.B. den Funktionsnamen dynamisch zu setzen), hierzu ein Beispiel:

```
stmt = "CALL" Funktname
INTERPRET stmt
```

ITERATE – Schleife nochmals durchlaufen

Die Iterate-Anweisung wiederholt einen Schleifendurchlauf:

```
DO i = 1 TO 10
  IF I % 2 = 0 THEN ITERATE /* gerade Durchlaufe überspringen */
 ...
END
```

LEAVE – Schleife verlassen

Die Leave-Anweisung beendet eine Schleife, zum Beispiel:

```
DO i = 1 TO 10
  IF I > 5 THEN LEAVE    /* 5. bis 10.-Durchlauf überspringen */
 ...
END
```

NOP – keine Operation

Die NOP-Anweisung führt keine Verarbeitung durch. Sie dient als Platzhalter in bedingten Anweisungen:

```
IF a = b
  THEN NOP
  ELSE ...
```

NUMERIC – numerische Formate und Genauigkeit bestimmen

Die Anweisung »NUMERIC« bestimmt die numerischen Formate und die Genauigkeit von Berechnungen. Obwohl ReXX standardmäßig mit einer Genauigkeit von 9 Ziffern rechnet, ist die maximale Genauigkeit praktisch unbegrenzt (aber zeitaufwendig!). Die Numeric-Anweisung kann auch verwendet werden, um das Format von numerischen Werten zu bestimmen.

```
NUMERIC DIGITS 32              /* 32-Ziffern Genauigkeit */
```

PARSE – Daten analysieren

Die Parse-Anweisung analysiert Daten und weist die Ergebnisse Variablen oder Platzhaltern zu (ein Punkt wird als Platzhalter verwendet). Die Anweisung »Parse« ist der mächtigste ReXX-Befehl.

ReXX bietet die folgenden Analysenarten:

– nach Position (absolut oder relativ)

– nach Zeichenfolgen

– nach Wörtern (ReXX-Wörter sind Zeichenfolgen, die durch Leerzeichen getrennt sind)

Eine Parse-Anweisung kann die verschiedenen Analysenarten enthalten:

```
liste = "Fehlermeldung 1; ohne Erklärung "

PARSE VAR liste wort1 wort2      /* wort1 = Fehlermeldung   */
PARSE VAR liste str 7 .          /* str = Fehler            */
PARSE VAR liste . ";" wort .     /* wort = ohne             */

PARSE VAR liste str1 7 str2 +3 . /* str2 = mel              */

such = "ung"
PARSE VAR liste str (such) .     /* str = Fehlermeld        */

such = "7"
PARSE VAR liste str =(such) .    /* str = Fehler            */

PARSE VAR liste str1 11 such +3 str2 (such) . ,
                                /* str2 = "1; ohne Erklär"  */
```

PROCEDURE – Unterprogrammprozedur

Die Anweisung »Procedure« identifiziert eine Routine mit versteckten Variablen. Variablen, die außerhalb der Routine zur Verfügung stehen sollen, müssen explizit mit dem Schlüsselwort »Expose« bekannt gegeben werden:

```
m = 1; n = 2
CALL Funkt
SAY m n          /* 1 4 */
...
Funkt: PROCEDURE EXPOSE n
  m = 3
  n = 4
  RETURN
```

PULL – Erstes Element der Queue holen

Die Anweisung »Pull« holt das erste Element aus der aktuellen ReXX-Queue. Die Pull-Anweisung ist eine Untermenge der Anweisung »Parse Pull«. Ist die Queue leer, wird die Eingabe von der Tastatur geholt. Es wird also auf eine Terminaleingabe gewartet.

```
PULL elem
```

PUSH – Element an den Anfang der Queue stellen

Die Push-Anweisung stellt ein Element an den Anfang der aktuellen ReXX-Queue.

```
PUSH "Zeile 1"
PUSH "Zeile 2"
PULL elem /* Zeile 2 */
```

QUEUE – Element an das Ende der Queue stellen

Die Queue-Anweisung stellt ein Element am Ende der aktuellen ReXX-Queue>.

```
QUEUE "Zeile 1"
QUEUE "Zeile 2"
PULL elem          /* Zeile 1 */
```

RETURN – Routine verlassen

Die Anweisung »Return« verläßt eine Routine. Ein Rückgabewert kann spezifiziert werden. Er steht einmal in der Variablen »Result« und als Funktionswert zur Verfügung.

```
CALL Funkt
SAY RESULT      /* 8 */
...
Funkt:

RETURN 8
```

SAY – Anzeige (Textausgabe)

Die Say-Anweisung gibt eine Meldung am Bildschirm aus:

```
SAY "meine Meldung"
```

SELECT...WHEN... OTHERWISE...END – Bedingte Selektion

Die Select...When...Otherwise...End-Anweisung führt die erste When-Klausel eines Select-Blocks aus, die die Bedingung erfüllt. Die Otherwise-Klausel wird ausgeführt, falls keine Bedingungen vorher erfüllt wurde.

Äquivalent zum Select- oder Case-Statement in C und Pascal wird nur eine Bedingung ausgeführt, wie das Beispiel verdeutlicht:

```
SELECT
  WHEN a > 0 THEN DO
    /* positiver Zweig */
    END
  WHEN a < 0 THEN DO
    /* negativer Zweig */
    END
  OTHERWISE
    /* sonstige Verarbeitung */
END
```

SIGNAL – Sprung

Die Anweisung »Signal« hat drei verschiedene Funktionen:

– Routine ansteuern (der Sprungname kann dynamisch gesetzt werden)

– Ausnahmebedingung aktivieren (z.B. Kommando-Fehler-Verarbeitungsroutine)

– Ausnahmebedingung deaktivieren

```
SIGNAL ON ERROR NAME CmdFehler
...
CmdFehler:
  SAY "Kommandofehler hat stattgefunden"
```

TRACE – Laufzeitinformationen anzeigen (Testhilfe)

Die Trace-Anweisung gibt Laufzeitinformationen am Bildschirm aus. Diese Anweisung kann interaktiv ablaufen, das heißt, der Anwender muß jede Ausgabe bestätigen (er kann jede ReXX-Anweisung interaktiv ausführen, auch »Trace« deaktivieren). Die Ausgabeinformationen können parametrisiert werden, beispielsweise mit »A« (alle Anweisungen), »E« (nur Kommandofehler) und »L« (durchlaufene Marken). Hierzu ein Beispiel:

```
TRACE ?A /* alle Anweisungen interaktiv (=?) anzeigen */
```

»=« – Zuweisung

Die »=«-Anweisung weist einer Variablen einen Wert zu:

```
a = 2
```

An dieser Stelle noch ein Hinweis:

```
a = b = 2
```

ist keine doppelte Zuweisung. Die zweite Operation mit »=« ist eine Vergleichsoperation, die zuerst ausgeführt wird. Als Ergebnis dieser Operation wird A zugewiesen, oder, falls B 2 ist, 0.

Befehle

Ein Befehl ist jede Anweisung, die keine Schlüsselwort-Instruktion ist. Ein Befehl wird an den aktuellen Kommandohandler übergeben. Die Adress-Anweisung setzt die aktuelle Befehls-Umgebung:

```
ADDRESS CMD
"DIR C:"
```

7.1.4 Standard-Funktionen

ReXX besitzt eine ganze Reihe von Standardfunktionen (sogenannte »Built-In«-Funktionen). Jede vollwertige ReXX-Implementierung enthält alle Standardfunktionen. Da die detaillierte Beschreibung der Standardfunktionen zu umfangreich wäre, wird hier nur ein grober Überblick gegeben.

Die ReXX-Standardfunktionen können in die folgenden Kategorien grob unterteilt werden:

– arithmetische (z.B. MAX)

– Bit-Bearbeitung (z.B. BITAND)

– Vergleich und Prüfung (z.B. VERIFY)

– Datenumwandlung (z.B. TRANSLATE)

– Datenerstellung (z.B. RANDOM)

– Zeit und Datum (z.B. DATE)

– Fehlerbehandlung (z.B. CONDITION)

– Format (z.B. FORMAT)

– Eingabe/Ausgabe (z.B. CHARIN)

– Zeichenkettenbearbeitung (z.B. REVERSE)

– Wortbearbeitung (z.B. WORDPOS)

Dazu einige Beispiele:

```
x = TRANSLATE("klein")                    /* -> KLEIN */
x = REVERSE("klein")                      /* -> nielk */
n = WORDINDEX("beta","alpha beta gamma delta")  /* n = 2 */
n = LENGTH(STRIP, "  alpha beta   "))          /* n = 10 */
```

7.1.5 Hilfsfunktionen (»REXXUTIL«)

Jede ReXX-Implementierung liefert eine Bibliothek mit Hilfsfunktionen passend für die entsprechende Umgebung. Solche Hilfsfunktionen stellen Funktionalitäten zur Verfügung, um ReXX-Aufgaben zu vereinfachen oder zum Teil überhaupt erst zu ermöglichen. Zum Beispiel werden in der OS/2-Umgebung Funktionen für die Workplace-Shell-Bearbeitung zur Verfügung gestellt. Obwohl viele hilfreiche Funktionen zur Verfügung stehen, decken sie nicht den vollen Umfang des Betriebssystems OS/2 ab.

Die OS/2-ReXX-Hilfsfunktionen werden in die folgenden Kategorien unterteilt:

– Profilvariablen ansprechen (z.B. SysIni)

– Meldungsbearbeitung (z.B. SysGetMessage)

– Workplace-Shell-Bearbeitung (z.B. SysCreateObject)

– erweiterte Attribute bearbeiten (z.B. SysGetEA)

– Dateien/Verzeichnisse bearbeiten (z.B. SysFileSearch)

Da diese Hilfsfunktionen keine Standardimplementierung von ReXX sind, müssen sie registriert werden.

```
/* REXXUTIL registrieren */
CALL RxFuncAdd "SysLoadFuncs", "RexxUtil", "SysLoadFuncs"
CALL SysLoadFuncs
SAY SysDriveMap()        /* angeschlossene Laufwerke holen */
```

7.1.6 REXXTRY – ReXX-Anweisungen testen

Der Befehl »RexxTry« wird eingesetzt, um eine ReXX-Anweisung interaktiv auszuführen. Das Ergebnis wird direkt am Bildschirm gezeigt. Die eingegebenen Anweisungen stehen im Zusammenhang, das heißt, ein lauffähiges Programm kann iterativ entwickelt werden. »Rexx-Try« kann für den neuen ReXX-Anwender sehr hilfreich sein, weil er Anweisungen sofort testen kann.

Der Aufruf erfolgt mit

```
REXXTRY
```

beziehungsweise

```
REXXTRY Anweisungen
```

Die REXXTRY-Anforderungsmaske erscheint (siehe Abb. 7.1).

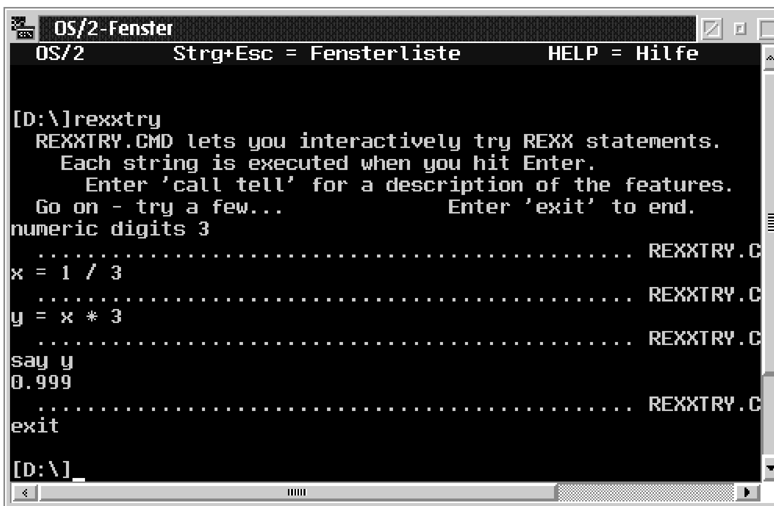


Abb. 7.1: Die Anforderungsmaske von REXXTRY

7.1.7 PMREXX – ReXX-Prozeduren als Gesamtbildschirm-Anwendung ausführen

Der PMREXX-Befehl führt eine ReXX-Prozedur als Gesamtbildschirm-Anwendung aus. Die Prozedur-Ausgabe wird in einer Maske dargestellt, die vorwärts und rückwärts gescrollt werden kann, das heißt, die Gesamtausgabe steht am Ende der Ausführung zur Verfügung. Der Name der auszuführenden Prozedur sowie die Aufrufparameter müssen spezifiziert werden.

```
RMREXX a:r
```

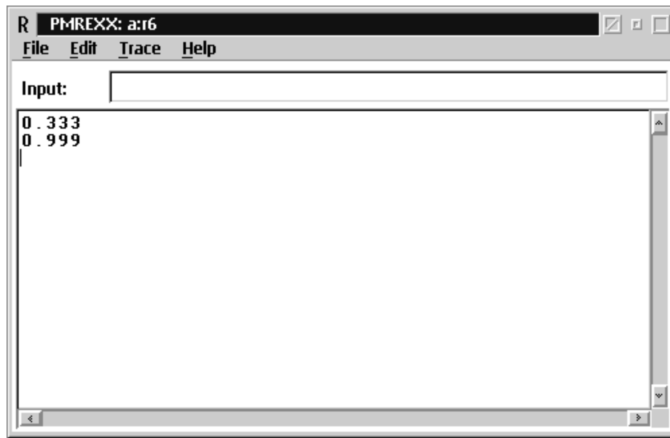


Abb. 7.2 Die PMREXX-Anforderungsmaske

7.1.8 Erweiterbarkeit

Ein wichtiges Ziel der Programmiersprache ReXX ist deren Erweiterbarkeit. Da die ReXX-Systemschnittstellen offengelegt sind, können andere Anwendungen ReXX-fähig gemacht werden. Zum Beispiel können sie, statt ihre eigene Skript-Sprache zu entwickeln, ReXX benutzen. Diese Fähigkeit wird in OS/2 fast universell verwendet, so daß ReXX in OS/2 fast ausschließlich als Skript-Sprache verwendet wird. Der Entwickler kann mit wenig Aufwand seine Anwendungen ReXX-fähig machen. Dies geschieht entweder mit einer Funktionsbibliothek oder mit einem Command-Handler, allerdings setzt dies Kenntnisse einer konventionellen Programmiersprache wie C voraus. Standard-Programmschnittstellen (sogenannte APIs) für die Verwendung von ReXX-Funktionalität stehen oft zur Verfügung.

7.1.9 Anwendungsgebiete

Wir haben jetzt die Basisfähigkeit von ReXX kennengelernt. Die Flexibilität von ReXX ermöglicht den Einsatz in vielen unterschiedlichen Bereichen:

– Skriptsprachen (z.B. für eine Installationsprozedur)

– Auswertungsprogramme (oft wird die längere Ausführungszeit durch die kürzere Entwicklungszeit mehr als wettgemacht).

– Dateiumformatierung (die mächtigen Textverarbeitungsfunktionen und die PARSE-Instruktion vereinfachen die Verarbeitung)

– Einzelanwendungen (viele Software-Produkte (z.B. VisPro/ReXX, VX-ReXX) verwenden ReXX als Programmiersprache.

– Ad-hoc-Programme

Ich werde im folgenden zwei Praxisbeispiele von lauffähigen ReXX-Programmen zeigen.

Beispiel 1 – Textanalyse:

Die Einzelwörter in einem Dokument werden gezählt und das am häufigsten verwendete Wort ausgegeben. Das Programm setzt voraus, daß die Variable »Text« den Analysetext enthält.

```
/* ReXX */
text = " ... "             /* Analyse-Zeichenfolge       */
Wortanzahl. = 0            /* initialisieren             */
zaehler = 0                /* Anzahl der Einzelwörter    */
DO WHILE text \= ""        /* Es ist noch Text vorhanden */
  PARSE VAR texte wort text
                      /* »wort« enthält die Einzelwörter */
  IF (Wortanzahl.wort = 0) THEN DO     /* neues Wort     */
    zaehler = zaehler+1     /* Anzahl der Wörter erhöhen */
    Woerterbuch.zaehler = wort         /* Wort speichern */
  END
  Wortanzahl.wort = Wortanzahl.wort+1  /* Worthäufigkeit */
END

maxwort = 0
DO i = 1 TO zaehler
  wort = Woerterbuch.i                 /* Wort holen     */
  IF Wortanzahl.wort > maxwort THEN DO /* Worthäufigkeit */
    maxwort = i
    haeufigkeit = Wortanzahl.wort
  END
END

SAY zaehler "Einzelwörter"
SAY  Woerterbuch.maxwort "kommt" haeufigkeit"-mal vor"
```

Beispiel 2 – Installationsprogramm:

Das Programm zeigt die Verwendung von einigen Hilfsfunktionen. Da ein DLL-Eintrag geladen wird, wird der LIBPATH-Eintrag aus »config.sys« gesucht, um die bestehenden Bibliotheksnamen zu finden. Zwei Angaben werden vom Anwender verlangt: die OS/2-Startpar-

tition, mit den Ini-Dateien von OS/2) und der Name der Anwendungs-DLL. Anschließend wird auf dem Arbeitsplatz eine Mappe mit dem Programmeintrag angelegt.

```
/* ReXX */
                                     /* REXXUTIL registrieren */
CALL RxFuncAdd 'SysLoadFuncs', 'RexxUtil', 'SysLoadFuncs'
CALL SysLoadFuncs
map = SysDriveMap(,'LOCAL')         /* lokale Laufwerke holen */

systemdrive = ''
DO WHILE map \= ''
  PARSE VAR map drive ':' map         /* Boot-Laufwerk suchen */
  filespec = STRIP(drive':\os2.ini')
  rc = SysFileTree(filespec,s.,'FST') /* Unterverzeichnisse   */
                       /* durchsuchen, Datum/Zeit zurückgeben */
  IF rc = 0 & s.0 > 0 THEN DO
     PARSE VAR s.1 jj '/' mm '/' tt '/' .
     systemdrive = systemdrive drive '('jj'/'mm'/'tt')' /*
                /* Laufwerk mit dem zuletzt verwendeten Datum */
  END
END
SAY "mögliche Systemlaufwerke:" systemdrive
PULL drive
                             /* LIBPATH aus CONFIG.SYS holen: */
rc = SysFileSearch("LIBPATH",drive":\CONFIG.SYS",s.)
IF rc > 0 THEN DO
  SAY "CONFIG.SYS nicht gefunden; Installation abgebrochen"
  EXIT
END
PARSE VAR s.1 'LIBPATH' '=' libpath    /* LIBPATH analysieren */
DO WHILE libpath \= ''
  PARSE VAR libpath dllname ';' libpath
  SAY dllname "; J = übernehmen, sonst ENTER"
  PULL eingabe
  IF eingabe = 'j' | eingabe = 'J' THEN LEAVE
END
SAY dllname "als DLL übernommen"
"COPY a:\rxddb2.dll" dllname"\*.*" /* System-Befehl für das
                                      Kopieren aufrufen       */

CALL SysCreateObject 'WPFolder','Test-Anwendung','<WP_DESKTOP>',,
                'OBJECTID=<USER_MAPPE>;','R' /* Mappe anlegen */
                                  /* Programmeintrag anlegen: */
setup = 'PROGTYPE=WINDOWABLEVIO;'||,
        'EXENAME=rxddb2.exe;'||,
        'PARAMETERS= [Anfrage eingeben];'
CALL SysCreateObject 'WPProgram','RXDDB2','<USER_MAPPE>',,
        setup,'R'
```

7.2 OpenGL

von Wolfgang Engel

7.2.1 Die Geschichte von OpenGL

Die amerikanische Firma Silicon Graphics Incorporated (SGI) brachte in den 80er Jahren Workstations auf den Markt, deren Grafikhardware in der Lage war, die Darstellung der Grafik teilweise selbst, ohne Hilfe des Prozessors, zu übernehmen. Die Entlastung des Prozessors führte dort zu einer wesentlich höheren Ausgabegeschwindigkeit.

Zur Programmierung dieser Workstations verwendete SGI die IRIS GL, wobei der Name IRIS für die Modellreihe der Workstations steht und die Buchstaben GL die Abkürzung für Graphics Language darstellen. Die IRIS GL wurde von zahlreichen Anbietern lizensiert, erlangte dort aber nicht dieselbe Bedeutung wie auf der SGI-Plattform. Daher schlossen sich einige Firmen zusammen und gründeten das »Architectural Review Board« (ARB), um die Open Graphics Language (OpenGL) zu entwickeln und zu spezifizieren. Dazu zählen derzeit DEC, IBM, Intel, Microsoft und SGI. Mittlerweile existiert für praktisch alle grafisch orientierten Betriebssyteme eine OpenGL-Implementierung, was wesentlichen Einfluß auf die Marktentwicklung haben wird. OpenGL könnte der Industriestandard der ausklingenden 90er Jahre werden.

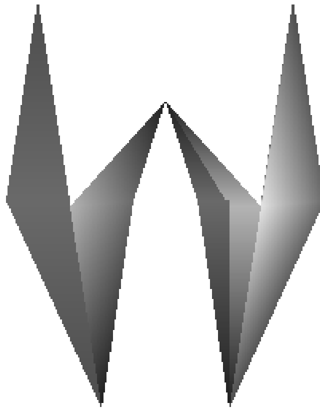


Abb. 7.3: Das Beispielprogramm in Aktion

7.2.2 Allgemeines zu OpenGL

OpenGL ist ganz auf maximale Geschwindigkeit bei der Darstellung von 3D-Objekten getrimmt. Es beherrscht keine Raytracing- und Radiosity-Rechnungen. Die OpenGL-API bietet mit rund 120 Funktionsaufrufen folgende Features:

– 3D-Ausgabe

– schattierte Punkte, Linien, Polygone

– Stacks für Attribute und Matrizen

– Ausblenden verdeckter Objekte (Depth Buffering)

– Arbeit mit Display-Listen

– Glätten von Linien- und Polygonkanten (Antialiasing)

– Subpixel-Rasterisierung

– Materialeigenschaften, komplexe Lichtmodelle, Texturmapping, atmosphärische Effekte

– verschieden Buffer für Spezialeffekte

– Netzwerktransparenz

Über diese Grundfunktionalität hinaus werden mit OpenGL oft zusätzliche Bibliotheken ausgeliefert, die OpenGL erweitern. Die OS/2-Implementierung enthält zwei dieser sogenannten Utility Libraries – AUX (Auxiliary Library) und GLUT (OpenGL Utility Toolkit). Beide Bibliotheken bieten einfache Windows- und Eingabefunktionen (zum Beispiel ein einfaches Popup-Menü). Außerdem erlauben sie

– eine gegenüber den OpenGL-Kernfunktionen vereinfachte Handhabung der geometrischen Transformationen

– die Darstellung von sogenannten High-Level-Primitiven wie Kugeln, Zylinder und ähnliches

– die Generierung von NURBS (Non Uniform Rational B-Splines)

– das »Aufbrechen« von komplexen Polygonen in Dreiecke (Polygon Tesselation)

– die Skalierung von Pixelbildern (Images)

– die Generierung von Texturen aus Images

Mit beiden Erweiterungsbibliotheken können betriebssystemunabhängig in kurzer Zeit kleine Programme erstellt werden, die dann mit wenigen Änderungen auf jedem Betriebssystem, das über eine OpenGL-Implementation verfügt, nach einem Re-Compile lauffähig sind. Sie eignen sich aber nur eingeschränkt zur Programmierung von komplexen und bedienerfreundlichen Anwendungen. Die Konzeption von OpenGL sieht nämlich eine weitgehende Betriebssystemunabhängigkeit vor. Es findet nur die eigentliche Darstellung, also das Rendering einer Szene mit Objekten (zwei- oder dreidimensional) statt. OpenGL verfügt über keine originären Interaktionsmechanismen. Um dem Anwender aber dennoch den Komfort des Look&Feel des jeweiligen Betriebssystems zu geben, stellt inzwischen fast jeder Betriebssystemhersteller eine Library zur Verfügung, mit der ein OpenGL-Programm in das jeweilige Betriebssystem eingebettet werden kann. Für OS/2 steht das OS/2-Presentation-Manager-GL-Interface (PGL) zur Verfügung.

7.2.3 Das GL-Interface des Presentation Manager

Das PGL wird über 17 Funktionsaufrufe gesteuert. Es beherrscht:

– Erzeugung und Verwalten eines OpenGL-Kontextes für das Rendering

– Austausch des Hintergrund- mit dem Vordergrundspeicherbereich, nach dem die Grafikinformationen in den Hintergrundspeicher geschrieben wurden

– Synchronisation

– Farbmanagement in einem Farbmodus mit Paletten

– Bindet sowohl das GPI- als auch OpenGL-Rendering ein, außerdem die Schriftdarstellung von Outline- und Bitmap-Fonts

Das kleinstmögliche PGL Programm durchläuft folgende Schritte:

– Test auf Lauffähigkeit: pglQueryCapability

– Die gewünschte OpenGL-Konfiguration anfordern: pglChooseConfig

– Erzeugung eines Kontextes: pglCreateContext

– Verbinden des OpenGL-Kontextes mit dem Fenster: pglMakeCurrent

– Rendering: allgemeine OpenGL-Funktionen

– Anzeige des Bildes: pglSwapBuffers

pglQueryCapability

testet die Lauffähigkeit der PGL-Anwendung auf dem Computer des Anwenders. Ein Aufruf von

```
rc = pglQueryCapability(hab);
```

liefert in *rc* einen Rückgabewert von NULL, 1, 2 oder 3. NULL zeigt an, daß OpenGL auf dieser Maschine nicht unterstützt wird. Wenn 1 zurückgegeben wird, ist eine Optimalkonfiguration vorgefunden worden, und es besteht eine OpenGL-Unterstützung durch den PM und DIVE. Demgegenüber beschränkt sich bei 2 diese nur auf den PM. Die 3 steht für eine erweiterte OpenGL-Unterstützung.

+

Mit *pglChooseConfig* kann eine Datenstruktur angefordert werden, die Aufschluß über die OpenGL-Fähigkeiten des jeweiligen Computersystems gibt. Dazu übergibt der Programmierer eine »Wunschliste«, für die dann von PGL die passende Grafikkonfiguration gesucht wird:

```
int attriblist [] =
{
 PGL_DOUBLEBUFFER,
 PGL_RGBA,
 None
};
```

Mit der hier beispielhaft gezeigten Attributliste wird ein Doppelpufferspeicherbereich angefordert. Außerdem sollen erstrangig nur RGBA-Farbkonfigurationen verwendet werden. Die gewünschte Konfiguration kann, muß aber nicht, einen oder mehrere *Auxiliary Color Buffer*, einen *Depth Buffer*, einen *Stencil Buffer* oder einen *Accumulation Buffer* haben. Das Wort *None* schließt die Struktur ab. Die Funktion gibt einen Zeiger auf eine *VISUALCONFIG*-Struktur – hier *vishead* – zurück:

```
vishead = pglChooseConfig(hab, attriblist);

typedef struct _VISUALCONFIG
{
  ULONG  vid;
  BOOL   rgba;
  int    redSize;
  int    greenSize;
  int    blueSize;
  int    alphaSize;
  ULONG  redMask;
  ULONG  greenMask;
  ULONG  blueMask;
  ULONG  accumRedSize;
  ULONG  accumGreenSize;
  ULONG  accumBlueSize;
  ULONG  accumAlphaSize;
  BOOL   doubleBuffer;
  BOOL   stereo;
  int    bufferSize;
  int    depthSize;
  int    stencilSize;
  int    auxBuffers;
  int    level;
  PVOID  reserved;
  struct visualconfig *next;
} VISUALCONFIG;
typedef  VISUALCONFIG *PVISUALCONFIG;
```

Die meisten Variablen sind selbsterklärend. Die Variable *vid* enthält eine sogenannte Visual-ID. Alle möglichen VISUALCONFIG-Konfigurationen eines Computers können mit *pglQueryConfigs* ermittelt werden. Mit *vid* können diese dann unterschieden werden. Die Variable *rgba* zeigt die Verfügbarkeit des RGBA-Modus an. RGB steht hier für Rot, Grün und Blau; *A* steht für *Alpha*. Beispielsweise erhält man mit

```
glColor4f(1.0, 1.0, 0.0, 0.5);
```

die Farbe Gelb und den Alphawert 0.5. Durch die Verwendung eines Alpha-Werts können verschiedene Effekte erzielt werden, am gebräuchlichsten dürfte jedoch die Interpretation als Transparenzwert sein. Mit 0.5 wäre das Objekt also halbdurchlässig. Zur Variable *auxBuffers*:

Eine OpenGL-Implementierung kann einen Speicherbereich anbieten, der praktisch zur freien Verfügung steht und keinen Beschränkungen unterworfen ist. Ein möglicher Zweck wäre bei-

spielsweise die Zwischenspeicherung von Images zur Senkung der Zugriffszeit. Das Bild müßte dann nicht jedesmal neu gezeichnet werden, sondern könnte während der Animation immer wieder ausgegeben werden. *auxBuffers* gibt die Größe dieses Speicherbereichs an.

IpglCreateContext

erzeugt dann den OpenGL-Kontext mittels der *VISUALCONFIG*-Struktur und gibt einen Handle – hier in der Variablen *hgc* – auf diesen Kontext zurück:

```
hgc = pglCreateContext(hab, vishead, (HGC) NULL, (BOOL) TRUE);
```

Die Übergabe von *TRUE* als letzten Wert zeigt an, daß die Ausgabefunktionen des GPI bei der Grafikausgabe umgangen werden sollen. Es soll also ein sogenannter »Direkter Kontext« verwendet werden, der über eine höhere Ausgabegeschwindigkeit verfügt.

pglMakeCurrent

verbindet einen OpenGL-Kontext mit einem OS/2-PM-Fenster. Es kann nur ein Kontext auf einmal an ein Fenster gebunden werden:

```
rc = pglMakeCurrent(hab, hgc, hwndClient);
```

War der Aufruf erfolgreich, steht in *rc TRUE*. Der erste Aufruf von *pglMakeCurrent* verbindet den Kontext mit der gesamten sichtbaren Fläche von *hwndClient*. Spätere Aufrufe ändern den Sichtbarkeitsbereich nicht mehr. Sollte dafür die Verbindung wieder gelöst werden, kann dies mit

```
rc = pglMakeCurrent(hab, NULL, None);
```

erfolgen. Um einen OpenGL-Kontext mit einem OS/2-PM-Fenster verbinden zu können, muß die Fensterklasse mindestens mit *CS_SIZEREDRAW* und *CS_MOVENOTIFY* erzeugt werden. Wichtig: Während der Zeit, in der ein Kontext mit dem Fenster verbunden ist, kann *WinSubclassWindow* nicht verwendet werden.

pglSwapBuffers

sorgt dafür, daß die Grafikinformationen angezeigt werden. Dazu muß dieser Funktion nur *hab* und der Handle des *PM-Fensters* übergeben werden:

```
pglSwapBuffers(hab, hwndClient);
```

Auf welchem Weg dann die Darstellung letztendlich erfolgt, hängt von den vom Programmierer angegebenen Werten der Attributliste ab und natürlich von deren Verfügbarkeit auf dem Zielsystem.

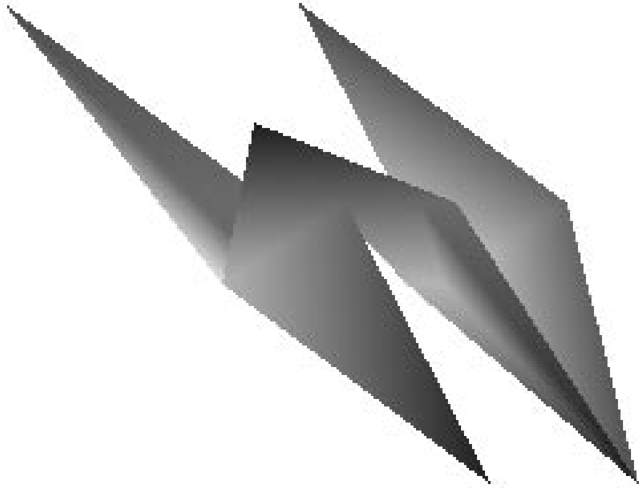


Abb. 7.4: Ein Screen des Beispielprogramms

7.2.4 Das Beispielprogramm

Das hier vorgestellte Beispielprogramm soll die Einbindung von OpenGL in OS/2 mit der PGL verdeutlichen. Dabei sollte das kleinst- und schnellstmögliche PGL-OpenGL-Programm entstehen. Die Ausgabe – und Renderingfunktion wurde in einen eigenen Thread gebettet. Die Abstimmung der Farben mit den OS/2-Systemfarben und die Anzeige des Bildes übernimmt vollständig die PGL. Zur Kompilierung des Beispielprogrammes wird das IBM Developer's Toolkit for OS/2 benötigt. Zu den dort vorgestellten zahlreichen Beispielen zur AUX- und GLUT-Library finden sich MAKEFILEs für das IBM C Set und für Visual Age C++. In den meisten Fällen muß eine *STACKSIZE* von mindestens 1 Mbyte verwendet werden. Um beispielsweise das OpenGL-Benchmark-Programm laufen lassen zu können, sind 4 Mbyte *STACKSIZE* notwendig. Alle dort aufgeführten Beispiele funktionieren auch mit Watcom C/C++ 10.00a und höher. Allerdings gibt der Compiler in der Version 10.00a eine Fehlermeldung im Zusammenhang mit der Aufrufkonvention APIENTRY aus. Eine mögliche Lösung ist hier ein Austausch von *APIENTRY* durch *_syscall* in den OpenGL-Headerdateien. Außerdem sollte bei Verwendung des Watcom-Compilers darauf geachtet werden, daß eine *Stack based calling convention* gewählt wurde. Dazu sollte die Einstellung *Pentium Stack based calling* in dem Einstellungsnotizbuch *C Compiler Switches* im *Memory Model and Processor Switches*-Dialog gewählt sein. Der entsprechende Kommandozeilenparameter für einen Pentium-Prozessor lautet *-5s*. Natürlich muß auch mit dem Watcom-Compiler ein entsprechend großer Stack gewählt werden. Eine Make-Datei »PMOpenGL.wpj« für die Watcom-IDE ab Version 10.00a liegt dem Beispielprogramm bei.

```
/* ==================================================================== */
/*                           PMOpenGL.C (gekürzt)                       */
/*        Beispiel für die Programmierung mit OpenGL und der PGL        */
/*                         (c) 1996 Wolfgang Engel                      */
/*          Das vollständige Listing finden Sie auf der CD-ROM          */
/* ==================================================================== */

#include [stdio.h]
#include "pgl.h"            /* Für die PGL */
#include "gl.h"             /* Für OpenGL  */
#include "PMOpenGL.H"

#define RenderWidth   400   /* Größe des Render-Fensters */
#define RenderHeight  400
#define kFramesToTime  16   /* Intervallänge der Zeitabfrage */
#define SQRT2       1.414   /* ... eine schnelle Wurzelberechnung von 2 */

HWND   hwndClient;
HWND   hwndFrame;
HAB    hab;
ULONG  ulToEnd        = 0;                /* Beendet while-Schleife */
ULONG  ulFramesToTime = kFramesToTime;
ULONG  ulNumFrames    = 0;                /* Framezähler      */
static float turnleftright = 0.0;         /* für die Bewegung */
static float turnupdown    = 0.0;

float colors[][3] = {                     /* zwei Farbmuster */
   {0.0, 1.0, 1.0},
   {0.0, 1.0, 0.0},
   {0.0, 1.0, 0.0},
   {0.0, 0.0, 1.0},
   {0.0, 0.0, 1.0},
   {0.0, 1.0, 1.0},
};

// gekürzt

int attriblist[] =                 /* Wunschliste für pglChooseConfig */
{
  PGL_DOUBLEBUFFER,                /* Doppelpuffer         */
  PGL_RGBA,                        /* RGBA Farbschema      */
  None                             /* ... immer zum Schluß */
};

MRESULT APIENTRY AboutDlgProc(HWND   hWnd,
                              ULONG  msg,
                              MPARAM mp1,
                              MPARAM mp2);
```

```
/* ==================================================================== */
/* Schnelle Keyboardabfrage                                             */
/* ==================================================================== */
void keyboard (void)
{
  // gekürzt
}

/* ==================================================================== */
/* Der Anzeigethread                                                    */
/* ==================================================================== */
VOID APIENTRY BlitThread(ULONG parm1)
{
  ULONG ulTime0, ulTime1; /* damit wird die verstrichene Zeit berechnet */
  CHAR  achFrameRate[32]; /* zur Anzeige der Geschwindigkeit in der     */
                          /* Titelleiste mit WinSetWindowText           */
  while (! ulToEnd) {
    /* zählt die angezeigten Bilder */
    if (! ulNumFrames++)
      DosQuerySysInfo(QSV_MS_COUNT, QSV_MS_COUNT, &ulTime0, 4L);

    /* der User hat eine Taste gedrückt ... ? */
    keyboard();

    /* Ab hier rendert OpenGL ...  */
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    /* Sechseck #1 */
    glPushMatrix();
    glRotatef(turnleftright, 0.0, 1.0, 0.0);
    glRotatef(turnupdown, 1.0, 0.0, 0.0);
    glBegin(GL_TRIANGLE_FAN);
      glColor3fv (colors[0]);
      // gekürzt
      glVertex3fv(verts [1]);
    glEnd();

// gekürzt

    glPopMatrix();

    /* Austausch von Hintergrund- und Vordergrundspeicher   */
    pglSwapBuffers(hab, hwndClient);

    /* alle sechzehn Bilder (ulFramesToTime == 16) wird die */
    /* Geschwindigkeit neu gemessen und angezeigt           */
    if (ulNumFrames >=ulFramesToTime) {
      DosQuerySysInfo(QSV_MS_COUNT, QSV_MS_COUNT, &ulTime1, 4L);
      ulTime1 -= ulTime0;
```

```
      if (ulTime1)
        sprintf(achFrameRate, "PM OpenGL %5.2f fps",
                (float) ((1000 <> 4) / (float) ulTime1));
      else
        sprintf(achFrameRate, "PM OpenGL %d fps", ulFramesToTime);

      WinPostMsg(hwndFrame, WM_COMMAND, (PVOID)ID_NEWTEXT, achFrameRate);

      ulNumFrames = 0;
    }
  }
  return;
}

/* ================================================================== */
/* Fehlermeldungen werden hier bearbeitet                             */
/* ================================================================== */
void WMB (PSZ errstr)
{
  // gekürzt
}

/* ================================================================== */
/* Setup von OpenGL                                                   */
/* ================================================================== */
void Setup()
{
  glColorMask(GL_TRUE, GL_TRUE, GL_TRUE, GL_FALSE);
  glDepthMask(GL_FALSE);
  glEnable(GL_CULL_FACE);
}

MRESULT EXPENTRY WindowProc(HWND hwnd, ULONG msg, MPARAM mp1, MPARAM mp2)
{
  // gekürzt
}

main(int argc, char **argv)
{
  PVISUALCONFIG vishead;                /* Visuelle Konfiguration */
  HMQ           hmq;                    /* Nachrichtenschlange    */
  QMSG          qmsg;                   /* Nachricht              */
  HGC           hgc;                    /* Der OpenGL-Kontext     */
  TID           tidBlitThread;
  ULONG         cxScreenSize, cyScreenSize, cxWidthBorder,
                cyWidthBorder, cyTitleBar, cyMenu;
  ULONG         createflags = FCF_TITLEBAR | FCF_SYSMENU    |
                              FCF_MINMAX   | FCF_SIZEBORDER |
                              FCF_ICON     | FCF_MENU;
  hab = WinInitialize(0);
```

```
hmq = WinCreateMsgQueue(hab, 0);
if (! hmq) WMB("Konnte die Message-Queue nicht erzeugen!\n");

/* Test auf OpenGL-Unterstützung. Rückgabewert:  */
/*      0 = kein OpenGL                          */
/*      1 = OpenGL mit PM und DIVE support       */
/*      2 = OpenGL nur mit  PM                   */
/*      3 = erweiterte Unterstützung von OpenGL  */
if (pglQueryCapability(hab)) {
   /* Gibt einen Zeiger auf die Minimalkonfiguration  */
   /* zurück, die den Ansprüchen des Programms genügt */
  vishead = pglChooseConfig(hab, attriblist);

  if (! vishead)
    WMB("Konnte keine passende visuelle Konfiguration finden\n");

  if (WinRegisterClass(hab, (PSZ) "PMOpenGL", WindowProc,
                       CS_SIZEREDRAW | CS_MOVENOTIFY,
                       /* Minimalanforderung */
                       0)) {
    hwndFrame = WinCreateStdWindow(HWND_DESKTOP,
                                   WS_VISIBLE,
                                   &createflags,
                                   (PSZ) "PMOpenGL",
                                   "PM OpenGL",
                                   WS_VISIBLE,
                                   0,
                                   ID_MAINWND,
                                   &hwndClient);
    if (! hwndFrame)  WMB("Konnte das Fenster nicht erzeugen!\n");

    /* Das Fenster zentrieren */
    cxWidthBorder=(LONG) WinQuerySysValue(HWND_DESKTOP,SV_CXSIZEBORDER);
    cyWidthBorder=(LONG) WinQuerySysValue(HWND_DESKTOP,SV_CYSIZEBORDER);
    cyTitleBar   =(LONG) WinQuerySysValue(HWND_DESKTOP,SV_CYTITLEBAR);
    cyMenu       =(LONG) WinQuerySysValue(HWND_DESKTOP,SV_CYMENU);

    cxScreenSize =(LONG) WinQuerySysValue(HWND_DESKTOP,SV_CXSCREEN);
    cyScreenSize =(LONG) WinQuerySysValue(HWND_DESKTOP,SV_CYSCREEN);

    WinSetWindowPos(hwndFrame, HWND_TOP,
                    cxScreenSize / 2 – ((RenderWidth+2 *
                                   cxWidthBorder) / 2),
                    cyScreenSize / 2 – ((RenderHeight+2 *
                                   cyWidthBorder+cyTitleBar+cyMenu)
                                   / 2),
                    RenderWidth  + 2 * cxWidthBorder,
                    RenderHeight + 2 * cyWidthBorder+cyTitleBar+cyMenu,
                    SWP_SIZE | SWP_MOVE | SWP_SHOW | SWP_ACTIVATE);
```

```
      hgc = pglCreateContext(hab, vishead,   /* Visuelle Konfiguration */
                             (HGC)NULL,
                             (BOOL)TRUE); /*direkter Kontext (schneller)*/

      if (! hgc) WMB("Konnte den OpenGL Kontext nicht erzeugen!\n");

      if (! pglMakeCurrent(hab, hgc, hwndClient))
      WMB("Konnte den OpenGL-Kontext nicht mit dem Fenster verbinden!\n");

      /* OpenGL setup */
      Setup();

      /* Eigener Ausgabethread */
      DosCreateThread(&tidBlitThread, (PFNTHREAD) BlitThread,
                      0L, 0L, 1000000L);

      /* Thread Priorität */
      DosSetPriority(PRTYS_THREAD, PRTYC_IDLETIME, 15, tidBlitThread);

      while (WinGetMsg(hab, &qmsg, NULLHANDLE, 0, 0))
        WinDispatchMsg(hab, &qmsg);

      /* beendet die while-Schleife */
      ulToEnd = 1;

      DosWaitThread ((&tidBlitThread), DCWW_WAIT);
    }
  }
}

/* ==================================================================== */
/* About-Dialog-Funktion                                                */
/* ==================================================================== */
MRESULT APIENTRY AboutDlgProc(HWND hWnd, ULONG msg, MPARAM mp1,MPARAM mp2)
{
  // gekürzt
}

/* ==================================================================== */
/*                        Ende von PMOpenGL.C                           */
```

7.3 OpenDoc, Componentware nicht nur für OS/2

von Christian Pürckhauer

Wir danken der Zeitschrift OS/2 Inside für die Genehmigung zum Abdruck dieses Beitrags, der bereits in den Ausgaben 2/96 bis 11/96 veröffentlicht wurde.

Am 21. März diesen Jahres hat die Object Mangement Gruppe (OMG) mit dem Document- und Data-Interchange-Facility einen weiteren wichtigen Standard der CORBA-Architektur verabschiedet. Der Vorschlag, Opendoc als Basis zu nehmen, wurde mit großer Mehrheit angenommen. Die Standards der OMG werden von einer wachsenden Anzahl von Ämtern und Industrieverbänden, wie z.B. des amerikanischen Gesundheitsministeriums, der Petroleumverbände, des US Verteidigungsministeriums, des MITREs und aller öffentlicher Ämter (FIPS), akzeptiert und sind Grundlage aller neuen Anwendungen. Damit besteht eine reale Chance, daß der wachsende Markt von Softwarekomponenten nicht allein auf proprietären Standards beruht, die von einem Hersteller kontrolliert werden, sondern sich neben OLE ein offener Standard etabliert, der auch aus technologischer Sicht viele Vorteile in sich birgt.

7.3.1 Componentware

Traditionelle Software hat die Tendenz, wie wir alle wissen, zu einem funktionsübeladenen Moloch zu wachsen. Die Anfälligkeit für Fehler wächst, der benötigte Plattenplatz steigt und steigt, ebenso die Zeit, die man auf ein Update wartet. Ein weiterer Nachteil ist, daß die Arbeit mit anderen Datentypen limitiert ist. Es gibt zwar meist eine Unzahl von Importfiltern und Konvertierungen, aber wenn neue Datentypen auftauchen, z.B. im Multimediabereich, so ist man doch mit herkömmlichen Anwendungen schnell am Ende seines Lateins. Der Gedanke von Komponenten geht davon aus, daß Software, die unabhängig voneinander entwickelt wurde, zur Laufzeit zusammengefügt werden kann. Einzelne Teile können ausgetauscht werden oder durch ein Update ersetzt werden, ohne Abhängigkeiten zu anderen Komponenten.

Komponenten sind Objekte mit einem Datenteil und einem Funktionsteil, um diese Daten zu bearbeiten. Dieser Teil wird Parthandler genannt. Komponenten, in Opendoc »Parts« genannt, können andere Komponenten, die über Drag&Drop eingebettet werden, ohne Einschränkungen des Datentyps aufnehmen. Dazu brauchen und ohne Wissen über die Art und Weise, wie diese Daten manipuliert werden. Dadurch ist dieses Modell offen für neue Datentypen, die z.B. aus dem WWW im Internet oder im Multimediabereich entstehen. Dies ist dem dem Plug&Play-Gedanken sehr ähnlich, wie wir ihn bei der Hardware kennen. Wenn wir eine neue Karte einbauen, erwarten wir, daß die Treiber dynamisch geladen werden, ohne daß das ganze System neu konfiguriert werden muß. Dabei ist es offensichtlich, daß die Schnittstellen standardisiert werden müssen. Wir können uns an einem Beispiel verdeutlichen, welche Aspekte und Schnittstellen eine Architektur für Software-Komponenten berücksichtigen muß. Alle Componentware-Architekturen, sei es OLE oder Opendoc, sind im Wesentlichen eine Sammlung von standardisierten Schnittstellen. Wir können uns die verschiedenen Aspekte anhand

eines einfachen Beispiels vor Augen führen, bei dem ein Grafik-Part über ein Text-Part gezogen und eingebettet wird.

Ein Dokument, das aus mehreren Parts zusammengesetzt ist, erscheint dem Benutzer gegenüber als eine konsistente Anwendung. D.h., die Parts, die unabhängig entwickelt wurden, müssen bei einer Reihe von Aspekten kooperieren, obwohl sie nichts über die Datentypen der anderen Parts wissen. Diese Aspekte wurden hier entsprechend der Opendoc-Architektur gegliedert.

Compound Document Services

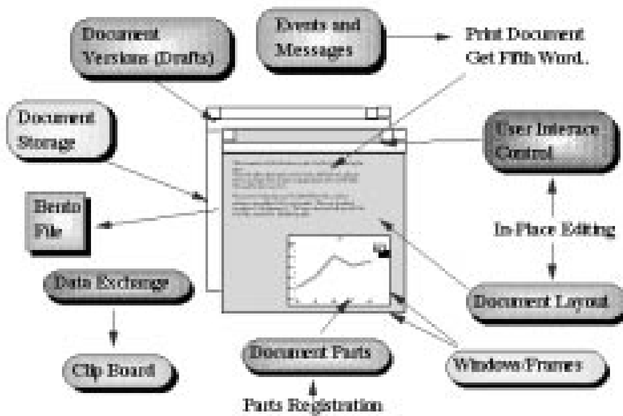


Abb. 7.5.: Was eine Component-Architektur enthalten muß

Text und Grafik sind unterschiedliche Parts mit unterschiedlichen Datentypen. Es gibt ein Set von Befehlen, wie z.B. beim Drucken des Dokuments, bei dem alle involvierten Teile in einer einheitlichen und definierten Weise reagieren müssen.

Beide Parts – Text und und Grafik – müssen kooperieren, wenn das Dokument editiert wird. Sie laufen in derselben OS/2-Anwendung. D.h., sie nutzen gemeinsam die Ressourcen dieses Prozesses. Nur ein Part, z.B. der Text-Teil, kann zur selben Zeit editiert werden. Dieser Part ist aktiv und hat den Focus. Dies bedeutet, er ist der Adressat aller Benutzeraktionen. Alle Tastatur-, Maus- oder Menü-Nachrichten werden an diesen Part geleitet.

Wenn der Focus geändert wird, weil z.B. der Grafik-Part aktiviert und editiert wird, muß der Text-Part, der vorher aktiviert war, seine Ressourcen wieder freigeben und die Kontrolle an den anderen Part übergeben. Der Grafik-Part seinerseits kann dann die Menüleiste modifizieren, so daß seine speziellen Aktionen angeboten werden.

Wenn das Dokument an eine andere Person verschickt wird, muß diese Person in Lage sein, mit diesem Dokument zu arbeiten, oder es zumindestens anzuschauen, auch wenn nicht dasselbe Set von Parthandlern auf dieser Maschine installiert ist.

Ein Part kann in der Lage sein, andere Parts aufzunehmen. Diese Parts werden Container-Parts genannt. Für andere Parts, z.B. einen Tickerpart, macht es eventuell keinen Sinn. Ein Container-Part muß in der Lage sein, mit eingebetteten Parts zu kommunizieren, um bestimmte Informationen zu erhalten und über ein gemeinsames Layout zu verhandeln. So wird ein Text sich vermutlich neu formatieren, um einen Grafikpart zu umfliessen. Eventuell muß eine neue Seite angelegt werden, um den restlichen Text aufzunehmen. Ein Text Part wird vermutlich anders reagieren, wenn der eingebettete Part ebenfalls ein Text-Part ist. Der Text wird dann warscheinlich in den vorhandenen Text eingebaut werden und nicht als separate Komponente eingebaut werden.

Die selben Entscheidungen müssen getroffen werden, wenn ein Part durch Clipboardaktionen wie Kopieren und Einfügen eingebettet werden soll.

Die Architektur muß eine gemeinsame Schnittstelle für das Layout eines Dokumentes definieren. Beide Teile müssen sich einigen über den Platz und die Koordinaten, die jedem Part beim Layout des Dokuments zur Verfügung stehen.

7.3.2 Speichersystem

Neben den Schnittstellen für ein Dokumentenlayout muß das Speichersystem standardisiert sein. Wenn ein Dokument, das aus unterschiedlichen Parts besteht, gespeichert wird, müssen die unterschiedlichen Datentypen in einer Datei gespeichert werden.

Wenn das Dokument, das unterschiedliche Datentypen in einer Datei gespeichert hat, wieder geöffnet wird, müssen die entsprechenden Parthandler den Daten wieder zugeordnet werden.

Wenn mehrere Leute gleichzeitig an einem Dokument arbeiten, muß es Mechanismen geben, die verhindern, daß die Änderungen einer Person durch das gleichzeitige Arbeiten einer anderen Person am selben Dokument zerstört werden. :eul.

Es muß standardisierte Protokolle zur Datenübertragung zwischen Komponenten geben. Dies sind Datenübertragungen durch:

– Drag und Drop innerhalb eines Dokuments oder zwischen mehreren Dokumenten,

– Clipboard-Aktionen wie Kopieren/Einfügen

– oder Links mit einer Datenquelle, die sich eventuell auf einer anderen Maschine befindet.

Scripting

Komponenten müssen benutzbar, anpaßbar und konfigurierbar sein, sowohl von Benutzerseite, als auch von anderen Parts.

Anstelle der anwendungsspezifischen Makro-Dialekte muß es einen einheitlichen Weg geben, Parts mittels einer Scriptsprache zu benutzen

7.3.3 Objekt-Modell

Wenn ein neues Release des Grafik-Parts verfügbar wird, muß es möglich sein, den einen Teil zu erneuern, dies aber ohne Abhängigkeiten zu anderen Parts. Dies erfordert ein standardisiertes binäres Interface der Komponenten (auch RRBC, Release To Release Binary Compatibility genannt). Erst diese Schnittstelle ermöglicht Plug&Play.

7.3.4 Alles über Parts

Ein Part ist der Basisbaustein eines Verbunddokumentes. Es besteht aus einem Datenteil und einem Parthandler, mit dem die Daten verändert und angezeigt werden können. Opendoc kennt darüber hinaus auch das Konzept eines Partviewers, mit dem die Daten eines Parts angezeigt und gedruckt werden können, also eine Teilfunktionalität eines Parthandlers. Der Grundgedanke dabei ist, Partviewer frei zur Verfügung zu stellen und nur die Parthandler zu lizenzieren.



Abb. 7.6: Parts

Parts sind unabhängige funktionale Einheiten, ähnlich einem ausführbaren EXE-Programm, aber mit folgenden Unterschieden

- Parts werden umgewandelt als DLLs und nicht als EXE-Dateien.
- Parts werden dynamisch geladen.
- Parts sind aufgaben- und datentypspezifisch. Anwendungen werden so zu Sets von konfigurierten zusammengesetzten Parts.

Ein Dokument besteht aus einem oder mehreren Parts: einem sogenannten »Root Part« und anderen Parts, die darin eingebettet sind. Benutzer stellen ein Dokument zusammen, indem sie, nachdem sie das grundlegende Layout-Modell eines Root-Parts gewählt haben, die einzelnen Bestandteile aus dem Part »Schablonen-Ordner« via Drag&Drop zusammenbauen. Diese Parts können auf die Arbeitsoberfläche oder in Dokumente gezogen und eingebettet werden.

Der Root-Part definiert die dokumentenweiten Optionen, z.B., ob ein Dokument automatisch oder nicht gesichert werden soll oder wie es gedruckt wird. Ein Dokument kann mehrere Ansichten bzw. Views haben. So kann die Hierarchie der eingebetteten Parts auch als Treeview angezeigt werden.

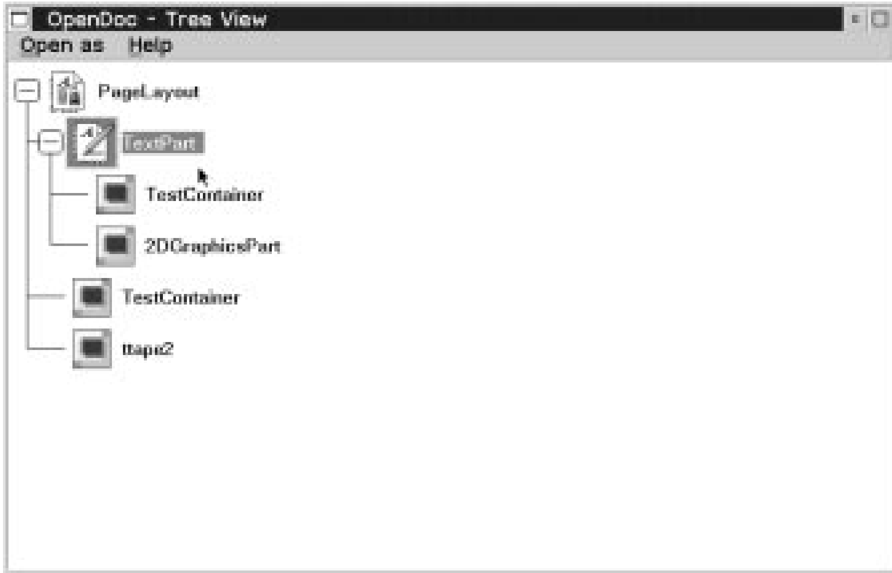


Abb. 7.7: Part-Hierarchie

In diesem Beispiel hat eine Textkomponente, eingebettet in ein Page-Layout-Part, zwei andere Parts eingebettet. Ob ein Part andere Parts einbetten kann oder nicht, ist eine Design-Entscheidung bei der Implementierung. Man nennt Parts, die andere Parts aufnehmen können, Container-Parts.

OpenDoc-Frames

Die logische Struktur, die einem Dokument zugrunde liegt, ist charakterisiert durch die Hierarchie, in der die Parts eingebettet und dargestellt sind. Ein Frame ist dabei der Bereich des Parts der visuell als Bereich dargestellt wird. Ähnlich wie Icons gibt es Frames-Handles zu den entsprechenden Parts, die es erlauben, Parts als Ganzes zu manipulieren. Mit Hilfe dieser Frames lassen sich Parts auf die Arbeitsoberfläche ziehen oder zwischen Fenstern bewegen. Sie können auch selbst Ziel für andere Frames bei Drag&Drop-Operationen sein. Neben der Editiermöglichkeit im Dokument selber (in Place Editing), lassen sich Frames auch als eigenes Fenster öffnen und wieder schließen, wenn z.B. der Platz im Layout des Dokumentes keine konfortable Bearbeitung zuläßt.

Im Gegensatz zu Icons erlaubt es ein Frame, daß der Inhalt angezeigt und bearbeitet werden kann. Ein Part kann dabei mehrere Frames geöffnet haben, um z.B. bei einem 3D-Part den Inhalt aus unterschiedlichen Perspektiven anzuzeigen oder in einer Kalkulation den Inhalt tabellarisch und grafisch darzustellen. Der Darstellungsmodus wird vom umgebenden Container gespeichert. So zeigt ein Folder Parts meist in der Form von Icons an. Wird dieser Part

dann in ein Verbunddokument gezogen, zeigt sich der Part als Frame. Ein Container betrachtet Parts als reguläre Teile seines Inhaltes, die unabhängig vom Inhalt selektiert, gelöscht, kopiert und bewegt werden können. Der eingebettete Teil ist dafür verantwortlich, sich zu zeichnen und die ihn betreffenden Nachrichten (Events) zu bearbeiten.

Wenn ein Part in einen anderen eingebettet wird, finden sogenannte »Frame Negotiations« statt, dies sind Verhandlungen über das gemeinsame Layout. Der eingebettete Teil möchte unter Umständen mehr Platz, als ihm der Container zugestehen kann, weil z.B. der Rand oder die Seite überschritten wird. Dies kann dazu führen, daß eine neue Seite angelgt werden muß. Das letzte Wort hat dabei immer der Container. OpenDoc stellt hierfür ein Set von APIs zur Verfügung.

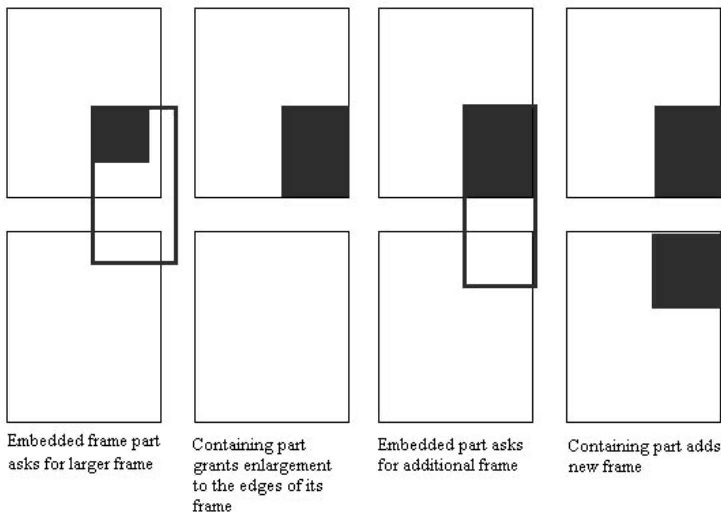


Abb. 7.8: Frame Negotiation

7.3.5 Attribute von Parts

Alle Parts haben ein gemeinsames Basisset von Attributen: der Typ des Parts mit dem der Parteditor wieder zugeordnet werden kann, eine übergeordnete Part-Kategorie (z.B. Text oder Video), die zur Suche alternativer Editoren dienen kann, der zuletzt benutzte Parteditor, der Anzeigemodus und so weiter. Diese Standardattribute können durch eigene selbstdefinierte Attribute erweitert werden. Die Partkategorien werden von CILabs verwaltet. Andere Parteditoren derselben Kategorie können Datenformate konvertieren.

7.3.6 OpenDoc Compound Document Services

In den Compound Document Services werden Schnittstellen beschrieben, die für das gesamte Dokument relevant sind. Sie beschreiben, wie beim Öffnen eines Dokuments z.B. der richtige Parthandler den jeweiligen Parts zugeordnet wird. Sie regeln außerdem die Verteilung von Nachrichten wie Benutzerinteraktionen (Maus-Events, Tastatureingaben etc.) an die jeweiligen Parthandler. Ein OpenDoc-Dokument läuft als eine OS/2-Opendoc-Anwendung in einem OS/2-Prozeß. Dieser Opendoc-Prozeß wird »DocShell« genannt und ist dafür verantwortlich, alle Aktionen und Operationen wie zum Beispiel Drucken zu steuern, die über die Parts eines Dokuments hinweg ausgeführt werden.

Nachrichten und Schlichtung von Konflikten

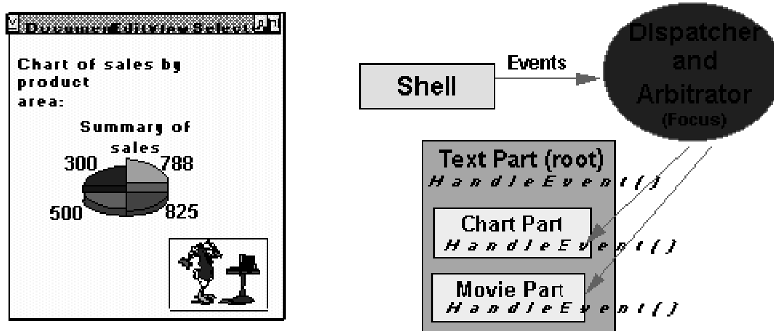


Abb. 7.9: OpenDoc-Nachrichten und Schlichtung von Konflikten

Die DocShell initialisiert das vom Dokument benötigte OpenDoc-Runtime-Environment. Dies geschieht, indem eine Instanz der Klasse ODSession instanziiert wird. Dieses Objekt instantiiert dann seinerseits alle wichtigen sessionweite Objekte wie das Speichersystem und speichert die Referenzen zu diesen Objekten. Will man als Entwickler auf diese Objekte zugreifen, so läßt man sich den entsprechenden Pointer auf das Objekt vom ODSession-Objekt geben.

Die DOC-Shell

Die DocShell öffnet die Datei und liest das Dokument in den Hauptspeicher, lädt bzw. bindet den zugehörigen Parthandler und zeigt das Window mit dem Dokument entsprechend der Statusinformationen an, die im ODWindowState-Objekt gespeichert sind. Es ist der Adressat für den OS/2-Presentation-Manager für alle Nachrichten. Die Docshell ihrerseits verteilt die Events dann weiter an die richtigen Parthandler mit Hilfe eines speziellen Objektes »ODDispatcher«.

– Er steuert alle dokumentenweiten Menü-Aktionen wie Drucken

– Bietet die Infrastruktur um Ressourcen wie Tastatur, Maus und Speicher gemeinsam zu nutzen.

– Erstellt neue Dokumente aus Schablonen

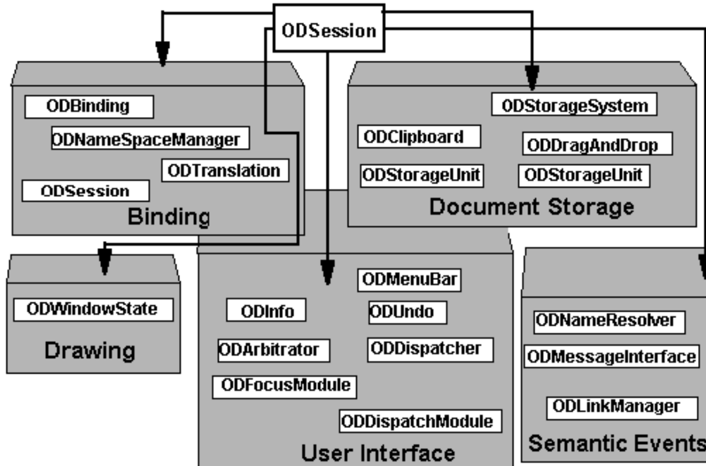


Abb. 7.10: OpenDoc-Session-Objekt

7.3.7 Binding

Wie bei den meisten Objekten wird lediglich der Datenteil gespeichert, wenn das Dokument gesichert wird. Die Daten werden im OpenDoc-Speichersystem »Bento« abgelegt. Ein Parthandler hingegen wird nur einmal geladen, auch wenn mehrere Parts derselben Art in einem Dokument vorhanden sind. Deshalb gibt es einen definierten Binding-Prozeß, mit dem Parthandler den Parts zugeordnet werden, wenn das Dokument geöffnet wird.

Diese Zuordnungen können in einem Notebook »Part Editor Preferences«, das sich im OpenDoc-Folder befindet, definiert werden. Diese Assoziationen für Part-Editoren können sowohl für Parttypen, die ein festgelegtes Datenformat haben, sowie für Part-Kategorien (z.B. Text) definiert werden. Opendoc hat eine Sequenz von 6 Stufen für diesen Binding-Prozeß. Dabei wird eine spezielle Serviceklasse »ODBinding« genutzt, um die richtige Entscheidung zu finden. In OpenDoc wird eine Tabelle verwaltet, in der Datentypen Parthandlern zugeordnet werden. Diese Zuordnung wird von einer anderen Serviceobjekt »ODNamespace« vorgenommen. Der Binding-Prozeß läuft rekursiv ab und startet mit dem äußersten Containerpart, der dann das Binding für die in ihm eingebetteten Parts aufruft. Zuerst wird nach einem Parthandler geschaut der als »prefered Editor« für diesen Typ definiert wurde

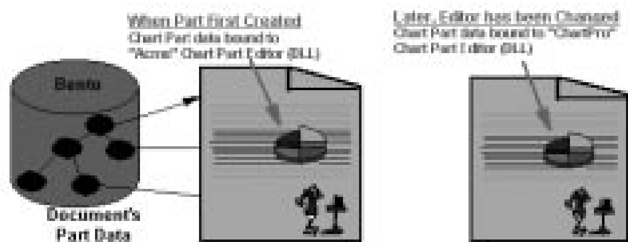


Abb. 7.11: Binding

Der zweite Schritt besteht darin, nach dem Editor zu suchen, mit dem der Part erstellt wurde. Im dritten Schritt wird nach einem beliebigen Parteditor gesucht, der diesen Parttyp unterstützt. Im vierten Schritt wird nach einem Parteditor derselben Kategorie gesucht, der mit dem Datenformat umgehen kann. Im 5. Schritt wird nach Datentransformationen gesucht, die von einem speziellen Serviceobjekt »ODTranslation« unterstützt werden und deren umgewandeltes Format von einem Editor unterstützt wird. Wenn nichts von alledem verfügbar ist, wird der sogenannte »Editor of last resort« gebunden, der den Teil als schwarze Box anzeigt, so daß der Benutzer mit dem Rest des Dokuments arbeiten kann.

7.3.8 Der Speicher von Open Doc

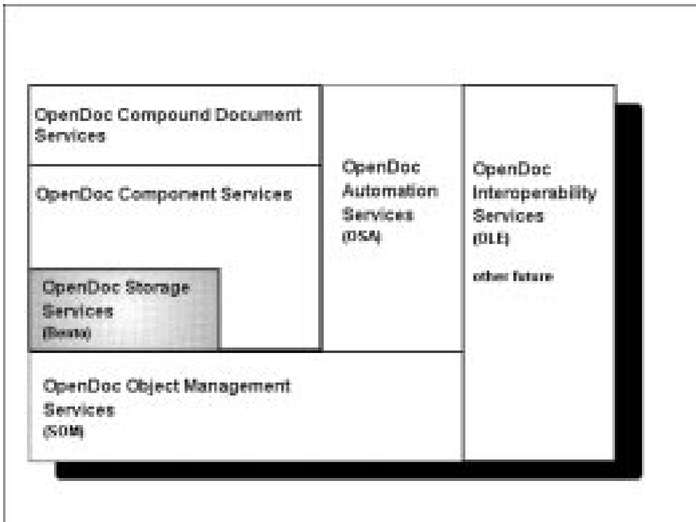


Abb. 7.12: Storage Services »Bento«

Ein OpenDoc-Dokument ist ein Container für die unterschiedlichsten Datentypen, wie Text, Video, Grafik etc. Um das Dokument als Einheit z.B. verschicken zu können, müssen diese unterschiedlichen Datenströme zusammen in einer Datei gespeichert werden können. OpenDoc bietet deshalb ähnlich wie bei OLE2 ein Speichersystem an, das in kleines Filesystem innerhalb einer Datei bietet. Das Speichersystem »Bento« wurde ursprünglich von Apple als effektiver dateibasierender Speicher für Multimedia-Dokumente entwickelt. Es ist nicht neu und wird von über 100 Produkten wie z.B.WordPro oder Lotus 1-2-3 seit Jahren verwendet. Bento wurde als Standard für Mulimedia-Container bei der Multimedia-Assoziation eingereicht.

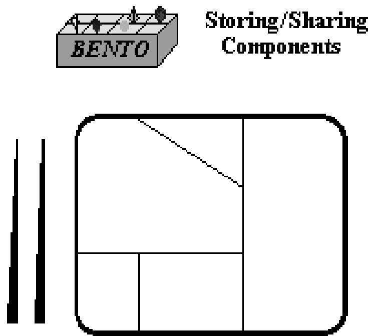


Abb. 7.13: Bento

Bento: *n.* [Japanese] 1. Abox lunch or picnic lunch. 2. A box or basket with multiple compartments, containing a collection of disparate elements arranged in an aesthetically pleasing manner.

Bento ist ein japanischer Begriff für eine Essensbox mit separaten Fächern für die unterschiedlichen Beilagen.

Das Speichersystem selbst ist über dem Filesystem des jeweiligen Betriebssystems implementiert und deswegen sehr portabel.

Bento speichert drei unterschiedliche Arten von Daten.

– Es stellt jedem Part einen eigenen Datenstrom zur Verfügung, um die eigenen Daten in dem eigenen jeweiligen Format abzuspeichern. Der Inhalt kann auch gleichzeitig in unterschiedlichen Formaten abgelegt werden. (z.B. bei einem Text-Part in ASCII, RTF, SAM). So können während des Binding-Pozesses auch andere Parthandler assoziiert werden.

– Es kann stabile Referenzen zu anderen Datenströmen innerhalb derselben Datei oder anderer Dateien speichern. OpenDoc verwendet eine Kette von solchen Referenzen, um die Beziehungen von eingebetteten Parts eines Dokuments zu speichern.

– Es speichert Metadaten, also Informationen über die gespeicherten Daten, in einer sehr effektiven Directory-Struktur ab, die es erlaubt, die Daten sehr schnell zu lokalisieren und darauf zuzugreifen.

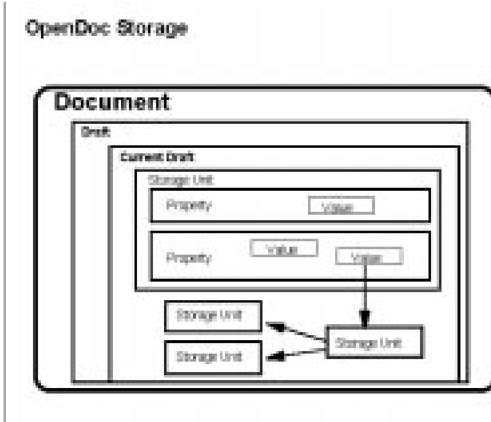


Abb. 7.14: Storage System Hierarchical Structure

Das Interface zum Speicher besteht aus einem System hierarchisch strukturierter Paare mit Eigenschaften und Werten (»property-value pairs« im OpenDoc-Slang). Die sogenannten »Properties« beschreiben dabei die Attribute der Daten, die »Values« die Werte. OpenDoc definiert ein Standard-Set von Eigenschaften, wie z.B. den bevorzugten Parteditor, den Autor, Zeit- und Datumseintragungen. Es können aber beliebige zusätzliche Eigenschaften definiert und gespeichert werden.

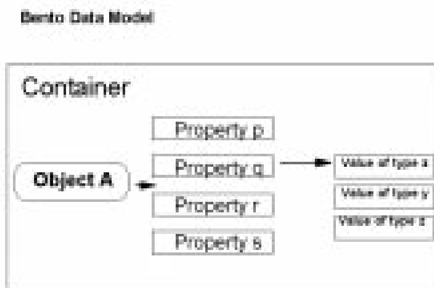


Abb. 7.15: Storage-Unit-Properties und Values

Die sogenannte Storage-Unit ist die Grundlage des Speichersystems. Es ist die Basiseinheit, mit der Daten transferiert werden. Es bietet ein Stream-Interface, in das die Daten geschrieben werden können, um entweder in einem OpenDoc-Dokument gespeichert zu werden oder um die Daten via Drag&Drop, Zwischenablage oder Linking zu übertragen. Die Klasse »ODStorageUnit« bietet so ein extrem mächtiges und in sich sehr konsistentes Set von Funktionen, um all diese Funktionen auszuführen. Es maskiert dadurch nicht nur die plattformabhängigen File-I/O-APIs, sondern auch die spezifischen APIs für Drag&Drop und Clipboard. Alle Storage-Units haben außerdem persistente IDs, die im Gegensatz zu OLE2 stabile Referenzen zwischen Datenbereichen auch in unterschiedlichen Dateien erlauben.

Das OpenDoc-Speichersystem ist eine abstrakte Schnittstelle, die über dem eigentlichen Speichersystem liegt, das mit Bento implementiert wurde. Das API ist aber eigentlich unabhängig von der Implementierung, so daß auch andere Implementierungen denkbar sind.

Entwickler können einen kurzn Blick auf das Beispiel unten werfen, um zu sehen, wie die Storage-Unit verwendet wird. Die ersten etwas verwirrenden Zeilen jeder Funktion sind die vom SOM-Compiler aus der IDL generierten Funktionsskeletons. Der Rest (mit // kommentiert) wurde von einem kleinen Tool mit Wizzards generiert, das mit dem OpenDoc-Toolkit mitgeliefert wird. Wenn ein Part neu initialisiert wird, wird einmalig die Methode »InitPart« aufgerufen, um den Part zu initialisieren. Dabei muß als erstes ein property-value-Paar vom Typ »kODPropContents« angelegt werden. Dies ist das Standard-Property, in dem die eigentlichen Daten des Parts abgelegt werden. Dann wird in dem Beispiel noch der bevorzugte Editor für den Part festgelegt. Die Methode *InitPartFromStorage* wird aufgerufen, um die Daten beim Öffnen des Dokuments wieder in den Speicher zu lesen. Die *Externalize*-Methode wird verwendet um die Daten zurückzuschreiben. Dabei wird wird zuerst die Methode *Focus* der Klasse ODStorageUnit aufgerufen, um den richtigen Datenbereich für den Part innerhalb der Datei zu lokalisieren.

Hier exemplarisch eine CPP-Datei, die vom Partmeister generiert wurde:

```
SOM_Scope void SOMLINK InitPart(CntrDrDropLink  *somSelf,
                                 Environment    *ev,
                                 ODStorageUnit* storageUnit,
                                 ODPart         *partWrapper)
{
  CntrDrDropLinkData *somThis = CntrDrDropLinkGetData(somSelf);
  CntrDrDropLinkMethodDebug(kPartHandlerName, "InitPart");
  CntrDrDropLink_parent_IBM_BaseNonContainer_InitPart(
                                               somSelf,
                                               ev,
                                               storageUnit,
                                              partWrapper);
  CommonInitPart(somSelf, ev);
 // store kind to storage unit
  storageUnit-AddProperty(ev, kODPropContents)->AddValue(ev,
                                                       kKind);
 // set preferred editor property to this part
  storageUnit->AddProperty(ev, kODPropPreferredKind)->AddValue(ev,
                                                       kODISOStr);
  StorageUnitSetValue(storageUnit, ev, strlen (kKind) + 1, kKind);
  _text = (string) SOMMalloc (strlen ("CntrDrDropLink") + 1);
  strcpy(_text, "CntrDrDropLink");
}

SOM_Scope void SOMLINK
              InitPartFromStorage(CntrDrDropLink *somSelf,
                                   Environment    *ev,
                                   ODStorageUnit* storageUnit,
                                   ODPart         *partWrapper)
{
```

```
  CntrDrDropLinkData *somThis = CntrDrDropLinkGetData(somSelf);
  CntrDrDropLinkMethodDebug(kPartHandlerName,
                            "InitPartFromStorage");
  CntrDrDropLink_parent_IBM_BaseNonContainer_InitPartFromStorage(
                                                   somSelf,
                                                   ev,
                                                   storageUnit,
                                                   partWrapper);
  CommonInitPart (somSelf, ev);

 // retrieve part data from persistent storage
  storageUnit->Focus(ev,
                     "Text",
                     kODPosUndefined,
                     kODKindPlainText,
                     0,
                     kODPosUndefined);
  long size = storageUnit->GetSize (ev);
  if (_text != NULL)
    SOMFree(_text);
  _text = (string) SOMMalloc(size);
  StorageUnitGetValue(storageUnit, ev, size, _text);
}
SOM_Scope void SOMLINK Externalize(CntrDrDropLink *somSelf,
                                   Environment    *ev)
{
  CntrDrDropLinkData *somThis = CntrDrDropLinkGetData(somSelf);
  CntrDrDropLinkMethodDebug(kPartHandlerName, "Externalize");
  ODStorageUnit* su = somSelf->GetStorageUnit (ev);
  if (su->Exists (ev, kODPropContents, kKind, 0))
  {
    su->Focus(ev, kODPropContents, kODPosUndefined, kKind, 0,
              kODPosUndefined);
    su->Remove(ev);
    su->AddValue(ev, kKind);
  } else
    su->AddProperty(ev, kODPropContents)->AddValue (ev, kKind);
  // store persistent data
  su->AddProperty(ev, "Text")->AddValue(ev, kODKindPlainText);
  StorageUnitSetValue(su, ev, strlen (_text) + 1, _text);
  CntrDrDropLink_parent_IBM_BaseNonContainer_Externalize(somSelf,
                                                        ev);
}
```

Draft

Einer der Vorteile des OpenDocs-Speichersystems ist, daß eine Versionsverwaltung unterstützt wird, die es erlaubt, unterschiedliche Stände eines Dokuments, sogenannte Drafts, im selben Dokument gleichzeitig zu speichern. Dabei wird jeweils nur die aktuellste Version komplett

als laufende Version gespeichert. Ansonsten wird nur das Delta zu den älteren Versionen gespeichert. Drafts beziehen sich immer auf das gesamte Dokument, nicht einzelne Parts. So können mehrere Personen am selben Dokument arbeiten ohne große zusätzliche Anforderungen an Plattenplatz.

7.3.9 Datenübertragung

OpenDoc bietet eine durchgängige Architektur für Datenübertragung, die auf derselben Storage-Unit basiert, die auch für die Speicherung in Dateien benutzt wird.

Drei Arten von Übertragung werden unterstützt: Drag&Drop, Clipboard und Linking.

Drag&Drop ist ein Mechanismus, der vom Prinzip dem Kopieren-/Ausschneiden-/Einfügen-Mechanismus sehr ähnlich ist, ohne dabei die Zwischenablage zu benutzen. Das Ziehen (Dragging), das einen Part verschiebt oder kopiert, wird auf unterschiedliche Art unterstützt. So kann bei großen Datenmengen nur eine Referenz (»promise« genannt) in der Storage-Unit abgelegt werden. Die Daten werden erst dann übertragen, wenn das Zielobjekt die Kopier- oder Verschiebeaktion akzeptiert. Wird ein Part kopiert, werden auch die eingebetteten Parts mit übertragen. Drag&Drop wird auch unterstützt zwischen OS/2-Ordnern und OpenDoc-Dokumenten.

7.3.10 Zwischenablage

OpenDoc unterstützt natürlich ebenfalls die Clipboardaktionen, die ausgehend von der Menüleiste über Kopieren, Ausschneiden und Einfügen angeboten werden, um Daten oder ganze Parts zu übertragen. Wie bei Drag&Drop wird die Übertragung innerhalb oder zwischen Dokumenten unterstützt. Dabei wird der Ausgangs-Part entweder als neuer Part in die Zielkomponente eingefügt oder die Daten werden in die Daten des Parts eingefügt (wenn z.B. ein Text-Part über einen anderen Text-Part gezogen wird).

7.3.11 Linking

Linking hält eine Verbindung zu der Datenquelle. Eine solche Datenquelle kann mit mehreren Zielparts verbunden sein, die über alle Änderungen der Datenquelle benachrichtigt werden. Die Zielobjekte können definieren, ob sie automatisch oder nur auf Anforderung über jede Änderung informiert werden wollen.

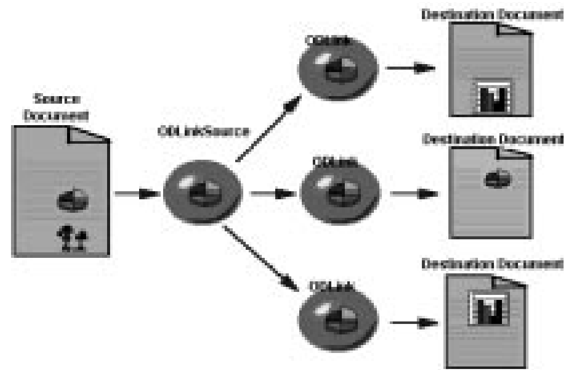


Abb. 7.16: Linking

Durch die Verwendung der Storage-Unit mit seinen persistenten IDs wurde ein sehr stabiler Linkingmechanismus für lokale und entfernte Objekte implementiert. Diese Links brechen nicht (im Vergleich zu OLE2, wenn die Zielobjekte verschoben werden. OLE2 referenziert Objekte mittels sogenannter »Monikers«, die typischerweise auf Filenamen basieren, verbunden mit einer Reihe weiterer Namen. Wenn die Datei verschoben oder umbenannt wird, brechen die Links. Ein weiterer Vorteil des Opendoc-Linking-Mechanismus ist es, daß Ziel- und Quell-Parts im Gegensatz zu OLE2 entkoppelt sind, so daß der Quell-Part nicht wissen braucht, welche Links alle aktiv sind.

7.3.12 Automation

Nun wollen wir uns dem Thema Scripting nähern. Es bietet fortgeschrittenen Benutzern die Möglichkeit, einfach Programme anzupassen bzw. zu erweitern und Abläufe zu automatisieren. Seit jeher stehen für Office-Anwendungen ein Set meist proprietärer Makroschnittstellen bereit, mit denen Sequenzen von Befehlen gestartet werden können. In den letzten Jahren wurden diese anwendungsspezifischen Makroschnittstellen durch unterschiedliche Basic-Dialekte wie Visual Basic, Lotusscript oder Star-Basic standardisiert, die wenigstens für die jeweils ensprechende Anwendungssuite eine einheitliche Scriptschnittstelle bietet: Visual Basic für MS-Office, Lotusscript für die Smartsuite und Star-Basic für StarOffice. Scripting boomt. Warum der neue Trend zum Scripting? Im Zuge von Network Centric Computing und dem Internet hat man die Möglichkeiten entdeckt, die Scripts bieten, wenn man sie im Netz betreibt. Dies wird häufig mit dem Begriff »mobile Agents« bezeichnet. Scripts werden interpretiert und laufen auf einer virtuellen Maschine, die die Unterschiede der Betriebssysteme kapselt. Scripts sind dadurch extrem portabel. Diese meist sehr kleinen Agent-Programme können im Netz wandern, Informationen sammeln, oder zum Beispiel Kaufaufträge durchführen, wenn der Preis ein bestimmtes Niveau hat und so weiter.

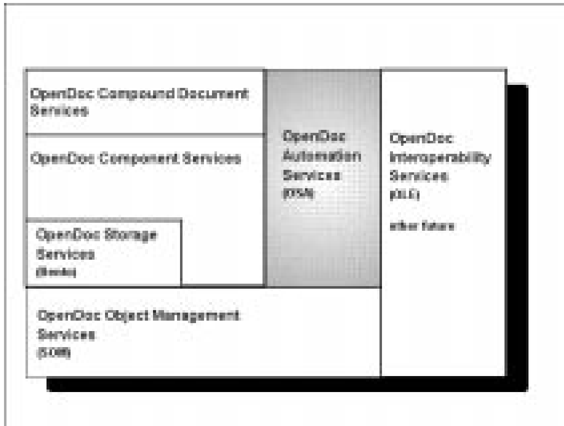


Abb. 7.17: Open Scripting Architecture

OpenDoc-Automation besteht aus drei Komponenten:

– *Open Scripting Architecture*, abgekürzt OSA genannt, die ursprünglich von Apple stammt, um eine Kommunikation und Zusammenarbeit unterschiedlicher Parts zu ermöglichen.

– *Semantische Nachrichten* (Events genannt), mit denen die Inhalte beschrieben werden können, an die Nachrichten adressiert sind.

– Erweiterungen, sogenannte *Extensions*, mit denen ein Part definieren kann, welche anwendungsspezifischen Sets von Nachrichten dieser Part unterstützt. OpenDoc standardisiert nur das Minimum-Set von Schnittstellen, die wichtig sind, damit Parts in einer Anwendung gemeinsam existieren können, wie Layout, Speicher, Ressourcen einer Session. Alle weitergehenden Formen von Kooperation sind Erweiterungen.

Die Open Scripting Architecture (OSA) definiert, wie Parts mittels Nachrichten, sogenannter »Events«, miteinander kooperieren können. Diese Nachrichen können entweder direkt aus einer Anwendung gesendet oder von einem Script generiert werden. Dieses Script kann dann z.B. mit einer Schaltfläche verknüpft sein, mit der das Script, und damit eine Sequenz von Befehlen, gestartet wird. OSA ist unabhängig von einer speziellen Scriptsprache. Eine Scriptsprache, die OSA unterstützt, muß aus den jeweiligen Scriptbefehlen beim Ablauf OSA Nachrichten generieren. Die Scriptsprache im OS/2 ist REXX. OS/2 bietet auch eine REXX-Schnittstelle für TCP/IP-Sockets und und FTP, so daß sich REXX hervoragend für Agenten- und Netzprogramme anbietet. (((In OS/2 Warp Version 4 kommt Object Rexx, ein Rexx, das nicht nur mit seinen objektorientierten Erweiterungen mittels SOM auf andere Objekte z.B. der Workplace Shell zugreifen kann, sondern auch für OSA angepaßt ist.))) Die Zusammenarbeit von Parts in OS/2 wurde realisiert mit einer Technologie die als »Interapplication Communication« IAC bezeichnet wird. Aus welchen Komponenten die IAC besteht, zeigt das nächste Bild



Abb. 7.18: Interapplication Communication

OSA erlaubt es, mehrere Anwendungen mittels eines Scripts zu steuern. Jede Scriptsprache hat eine entsprechende Script Komponente, die von einem »Component Manager« gesteuert wird. Wenn ein Script ausgeführt wird, sendet die Scripting-Komponente OSA-Nachrichten zu den Anwendungen, um die Aktionen, die im Script beschrieben, sind auszuführen. OSA ist in OS/2 Warp Version 4 unabhängig von OpenDoc implementiert und kann von allen Anwendungen benützt werden. OSA basiert auf Nachrichten. Es erweitert den bestehenden Nachrichtenmechanismus des Presentation Managers durch semantische Nachrichten. Das heißt, die Objekte, die sonst in einer Nachricht über einen Objekt- oder Windowhandle direkt adressiert werden müssen, können auf eine indirekte Art, nämlich semantisch, beschrieben und damit adressiert werden. Der Script Command: *copy table "Summary of Sales" to Totals* beschreibt die beiden Objekte Tabelle und Summe auf eine allgemeine Art.

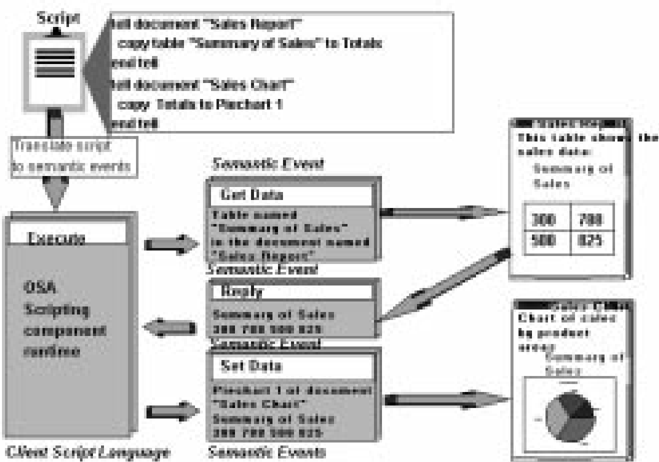


Abb. 7.19: Scripting

Die Scripting-Komponente übersetzt die für den Benutzer verständlichen Verben und Begriffe der Scriptsprache in OSA Nachrichten mit den entsprechenden Event IDs und Parametern. OSA unterstützt auch eine Aufnahmefunktion, die es Benutzern der Scripting-Komponente erlaubt, Aktionen und die dabei ablaufenden Nachrichten aufzunehmen und in Form eines Scripts abzuspeichern.

7.3.13 OSA-Events

OSA verschickt die Nachrichten mit Hilfe des Presentation Manager an andere Anwendungen. So kann eine Anwendung mit dem *Get Data*-Event von einer anderen Anwendung einen bestimmten Datenbereich, z.B. eine Tabelle, anfordern. Wird der Event von der anderen Anwendung unterstützt, so lokalisiert diese den Datenbereich und gibt eine Kopie davon zurück. Die anfordernde Client-Anwendung, zum Beispiel eine Scripting-Komponente, benutzt dabei Event-Manager-Routinen, um die Nachrichten zu erstellen und zu versenden. Wenn der OSA-Event eine Standardnachricht ist, die in der *OSA Event Registry: Standard Suites* registriert ist, kann die Nachricht sehr leicht von der Clientanwendung konstruiert und von der Serveranwendung interpretiert werden. Um sicherzustellen, daß eine Anwendung auf die entsprechenden OSA-Events reagieren kann, sollten die Standard-OSA-Nachrichten für die Anwendung unterstützt werden.

Die Standardsuites beinhalten:

– Die *Required Suite* die aus vier OSA-Events besteht. Diese sind: *Open Application*, *Open Documents*, *Print Documents*, and *Quit Application*. Diese Events müssen von einer Anwendung, die OSA unterstützt, bearbeitet werden können.

– Die *Core Suite* besteht aus einem Set von Basisfunktionen wie *Get Data*, *Set Data*, *Move*, *Delete* und *Save*, die fast von allen Anwendungen für Kommunikationszwecke verwendet werden.

– Funktionsspezifische Suites, die aus Gruppen von funktional zusammengehörigen OSA-Events bestehen. Diese Gruppen bestehen aus Text, Spreadsheet oder Datenbank-Suiten. Die Anwendung kann dabei entscheiden, welche Funktionen sie unterstützt. Eine Textanwendung sollte natürlich die Text-Suite unterstützen und so weiter

Diese Suites (außer der »Required Suite«, die im übrigen von jedem Part erweitert bzw. modifiziert werden können, werden in OpenDoc »Extensions« (Erweiterungen) genannt. Die Suites, die von einem Parthandler unterstützt werden, müssen registriert und damit nach außen hin bekannt gemacht werden, so daß andere Parts abfragen können, welche zusätzlichen Funktionen ein Part unterstützt. Diese Sets von Nachrichten werden von CILabs standardisiert und verwaltet. Dies hat den Vorteil, daß Parts einer bestimmten Kategorie nicht nur portabel sind, sondern die Interoperabilität auch standardisiert ist. Parts können dadurch auch von CILabs zertifiziert werden, was Benutzern die Sicherheit gibt, daß gekaufte Parts auch kooperieren bzw. ausgetauscht werden können. Eine Clientpart oder Anwendung kann dabei einen anderen fragen, ob eine bestimmte Extension unterstützt wird. Es kann Zugriff zu diesen Interfaces requestieren und die entsprechenden Funktionen dieser Schnittstelle aufrufen. (Dies ist in etwa

zu vergleichen mit der IUnknown-Schnittstelle bei OLE, mit der man herausfinden kann, ob ein Interface unterstützt wird oder nicht).

Ein OSA-Event besteht aus den folgenden vier Teilen:

– Events, die die Funktionen definieren (z.B. Open, Close, Get, Set)

– Event-Objekten, die die adressierten Objekte beschreiben bzw. spezifizieren (z.B. Dokument, Abschnitt, Wort, Tabelle), auf die Aktionen ausgeführt werden können.

– Deskriptoren, mit denen die Objekte lokalisiert werden können, (wie z.B. last, first, third)

– Parametern (wie z.B. bold, blue)

Ein Event-Objekt kann alles sein, das von einer Anwendung unterstützt wird. Dies sind hierarchisch geschachtelte Event-Objekte. Fast alles, was von einem Benutzer differenziert und manipuliert werden kann, ist als OSA-Event-Objekt beschreibbar. Wenn ein solches OSA-Event-Objekt semantisch von einem Script angesprochen wird, muß die Anwendung erst diese Objekte lokalisieren, um dann die angeforderte Aktion durchführen zu können. Diese Funktionen, um die Ziel-Event-Objekte zu lokalisieren, werden »Object Accessor« Funktionen genannt. Diese werden von einem OpenDoc-Objekt *ODNameResolver* in einer rekursiven Weise von außen nach innen aufgerufen.

7.3.14 Rexx

Object Rexx ist eine Erweiterung des klassischen Rexx, d.h die bisherigen Funktionen und Instruktionen werden weiter unterstützt. Ergänzt wurde die Unterstützung für Klassen, Objekte und Methoden, Nachrichten und Mehrfachvererbung

Corba-konformer Zugriff zum OS/2-System-Object-Model

Object Rexx enthält ein Set von Basisklassen wie ALARM, CLASS, ARRAY, LIST, QUEUE, TABLE, SET, DIRECTORY, RELATION, BAG, MESSAGE, METHOD, MONITOR, STEM, STREAM, STRING, und SUPPLIER. Daneben gibt es einen ganz neuen wesentlich performateren Interpreter. Rexx Commands können zusammen mit den neuen OO Erweiterungen und den OSA basierenden Kommandos verwendet werden. Object Rexx interpretiert die OSA basierenden Kommandos und schickt sie an den Event Manager.

Zusammen mit der OpenDoc Runtime wird ein Script Editor (eine angepasstes Derivat des altbekannten EPM) mitgeliefert, mit dem Scripts ablaufen und aufgenommen werden können.

Rexx sendet eine Nachricht durch das Script, um eine Verbindung aufzubauen. Die Tilde »~« ist der Message-Operator. Links davon ist das Objekt *scriptableapp*, an das die Nachricht geschickt wird, rechts davon die Nachricht *connect*. In der Klammer sind die Parameter

```
app =.scriptableapp~connect('OpenDoc-c:\opendoc\mydoc.doc')
```

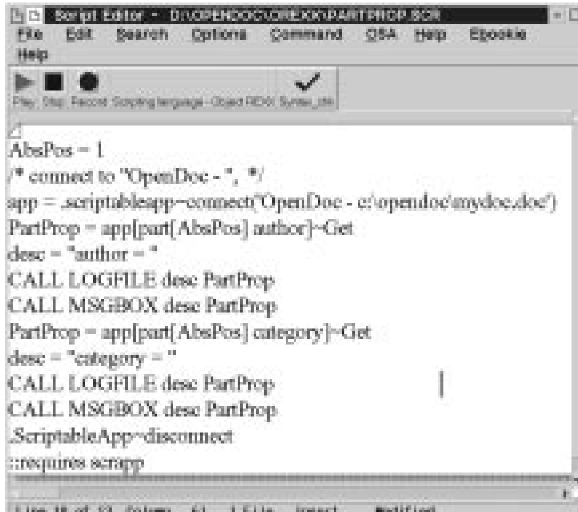


Abb. 7.20: Script-Editor

.scriptableapp ist ein Klassen-Objekt, das mit Object Rexx mitgeliefert wird, das die die Nachricht *connect* erhält, und eine Instanz dieser Klasse *scriptableapp*, das das spezifizierte OpenDoc Dokument-repräsentiert, zurückgibt. Ein Klassenobjekt ist ein Objekt, das in der Lage ist, Instanzen einer Klasse zu erstellen, in diesem Fall *app*. Klassenobjekte werden mit dem Präfix ».« dargestellt.

```
PartProp = app[part[AbsPos]author~Get]
```

Innerhalb des Dokuments wollen wir mit einem Part, den wir mit *AbsPos* als den ersten Part spezifiziert haben, kommunizieren. Dieser Part bekommt den Get-Data-Event zusammen mit einem zusätzlichen Parameter *Author* (ein Standard-Attribut) und soll den Wert dieses Attributs in der Variablen *PartProp* zurückgeben. Die Klassen, die von einem Script benötigt werden, sind in anderen CMD-Dateien definiert und können mit dem neuen Rexx-Statement *::requires* importiert werden. Dort ist im Beispiel das Interface der Klasse *scriptableapp* gespeichert.

7.3.15 Was kommt

OSA ist durch das Prinzip der semantischen Events ein sehr mächtiges Instrument. Ich möchte dabei nicht verschweigen, daß der Prozeß der Auflösung der adressierten Objekte heute auch Perfomanceprobleme mit sich bringen kann. Rexx auf der anderen Seite hat seine Liebhaber, ist aber für neue Benutzer sicher nicht so intuitiv zu erlernen wie andere, der natürlichen Sprache wesentlich näher angelehnte Scriptsprachen wie Apple-Script. OSA wird ergänzt werden durch einen nicht so konfortablen, dafür performanteren Skriptmechanismus, bei dem die Schnittstellen in SOM definiert werden und der ein problemloses Mapping von OLE-Automation-Strings ermöglicht.

7.3.16 Das Objekt-Modell

Wenn man Anwendungen aus Parts bauen will, müssen die einzelnen Objekte untereinander kompatibel sein. Das ist bei einem einzelnen Programm kein Problem, wenn man jedoch komplexere Client-Server-Systeme baut, müssen die Objekte kompatibel sein und untereinander interagieren können, auch, wenn sie in unterschiedlichen Programmiersprachen geschrieben wurden, auf unterschiedlichen Hardwareplattformen und unterschiedlichen Betriebssystemen laufen und über beliebige Netzwerkprotokolle miteinander kommunizieren.

Dafür benötigt man ein stabiles Objektmodell, das ein standardisiertes binäres Anwendungsinterface bietet, um Komponenten dynamisch zu laden und zu entladen.

– Ein Interface-Repository, das die entsprechenden Metadaten über die Objekte speichert. Dies ermöglicht es, unabhängig voneinander entwickelten Objekten, ihre Schnittstellen offenzulegen und es für andere Objekte zur Laufzeit abfragbar zu machen.

– Eine standardisierte Syntax und Semantik, um das Interface einer Klasse unabhängig vom Tool, mit der die Klasse implementiert wird, sprachneutral beschreiben zu können.

Objekttechnologie eignet sich grundsätzlich sehr gut, um solche Komponenten zu realsieren, da dort eines der wichtigsten Prinzipien, nämlich die Trennung von Interface und Implementierung, realisiert wird. Es gibt dort einen allgemein anerkannten Standard, den der »Object Management Group« OMG.

Die OMG ist ein Konsortium mit über 600 Mitgliedern. Es ist die einzige Standardisierungsgruppe im Bereich Objekttechnologie. Dort wurde eine Architektur verabschiedet, in der die gesamte Infrastruktur für Objekte definiert wird.

Kern dieser Object-Management-Architektur (OMA) ist eine Interface Definition Language, eine Standardisierung, wie Objekte beschrieben werden. Der erste große Baustein, der standardisiert wurde, war der »Object Request Broker«, auch CORBA genannt. Hier wird die Kommunikation zwischen Objekten in unterschiedlichen Prozessen und im Netz definiert. Teil dieser Architektur ist ein Interface Repository, in dem alle IDL-Definitionen der Klassen gespeichert sind und das es zur Laufzeit anderen Objekten ermöglicht, das Interface eines anderen Objekts zu erforschen.

Das »System Object Model« (SOM), das in OS/2 seit Version 2 implementiert ist, ist die Implementierung des CORBA-Objektmodells von IBM, das zusätzlich eine binäre Anwendungsschnittstelle für Objekte bietet, so daß eine »Release to Release Binary Compatibility« (RRBC) möglich ist. Dies ermöglicht außerdem, daß das Interface von Objekten sprachneutral ist. Das heißt, eine Klasse kann z.B. in C++ implementiert sein, eine davon abgeleitete Klasse in OOCobol oder mit einer nicht objektorientierten Sprache wie C, während z.B. Smalltalk- oder OORexx-Programme auf diese Objekte zugreifen können.

Dies heißt für OpenDoc-Komponenten, die ja auf SOM als Objektmodell basieren, daß sie Plug&Play unterstützen und so Komponenten ausgestauscht werden können, und dies ohne Abhängigkeiten zu anderen Komponenten. Das heißt aber auch, daß OpenDoc-Komponenten in unterschiedlichen Programmiersprachen implementiert werden können.

Durch SOM sind die OpenDoc-Komponenten CORBA konform und so steht ihnen die gesamte Welt der verteilten Objekte inklusive der gesamten Infrastruktur der Object-Services offen. Dies ist vielleicht der größte technische Vorsprung, den OpenDoc heute vor der OLE-Welt hat.

Bei der Definition der Objekt-Services ist die OMG der OLE-Welt um Jahre voraus. Dies wird es innerhalb eines sehr kurzen Zeitraums ermöglichen, diese Services auch von OpenDoc aus nutzbar zu machen. OpenDoc basiert heute noch auf Release 2.1x von SOM.

Neben dem eigentlichen Object Request Broker, der als Nachrichtenbus für die Kommunikation zwischen Objekten in einem Netz zuständig ist, standardisiert die OMG auch die gesamte C/S-Infrastrukur für Objekte in den sogenannten »Common Object Services« (COSS).

In SOM Release 3 für OS/2 wird bereits ein großes Subset dieser Services, wie *Naming*, *Transactions*, *Concurrency*, *Lifecycle*, *Events*, *Persistance* und *Externalization* bereitgestellt. Werden diese Services in Zukunft mit den OpenDoc-Parts verknüpft, steht eine sehr mächtige Infrastruktur für auch mission-kritische Geschäftsanwendungen zur Verfügung.

Object Services

Die wichtigsten Service ist der *Naming-Service*, mit dem Objekte mit verständlichen Namen in einem Netzwerk versehen und dort wieder lokalisiert werden können.

Transaction-Services erlauben Transaktionsverarbeitung mit verteilten Objekten, inklusive verschachtelter Transaktionen, two phase Commit und so weiter.

Concurrency Services definieren verschiedene Locksets, mit denen der gleichzeitige Zugriff auf Objekte geregelt bzw. serialisiert wird.

Lifecycle standardisiert das Erstellen und Löschen von verteilten Objekten in einem Netzwerk, inklusive Operationen wie *Copy* und *Move* über das Netz.

Mit den *Persistance Services* wird das Abspeichern von Objekten unabhängig vom darunterliegenden Datenspeicher standardisiert.

Externalisation definiert in Streaminterface für Objekte, so daß sie über das Netz geschickt werden können und auf der anderen Seite wieder Objekte daraus erstellt werden können.

Event Services ermöglichen einen asynchronen Nachrichtenaustausch.

Common Facilities und OpenDoc als Standard

Neben diesen Basis-Infrastrukurservices definiert die OMG in den CORBA-Facilities-Services Anwendungen, die direkt genutzt werden können. OpenDoc basiert auf einem CORBA-konformen Objektmodell. Dies war einer der entscheidenden Gründe, warum OpenDoc als Standard für die »Distributed Document Component Facilities« der OMG im März 1996 verabschiedet wurde.

OpenDoc ist implementiert als eine SOM-basierende Klassenbibliothek. Alle Parts sind entweder direkt oder indirekt von *ODPart* abgeleitet. Jeder neue Part muß mindestens eine hand-

voll Operationen überschreiben, in denen er definiert, wie sein Parthandler mit bereits gepeicherten Daten initialisiert wird, wie er seine Daten am Bildschirm anzeigt und wie die Daten abgespeichert werden. Alle wichtigen Klassen sind von *ODRefCount* abgeleitet, wodurch eine Garbage Collection unterstützt wird, durch die nicht mehr referenzierte Objekte au dem Hauptspeicher entfernt werden können.

Der Interoperabilitäts-Layer wurde ursprünglich von Novell entwickelt und wird zuerst mit der Windowsversion von OpenDoc ausgeliefert werden.

Dadurch können OLE2-Server als Parts in OpenDoc-Containern laufen und OpenDoc-Parts als OLE-Server in OLE-Containern. Erreicht wird dies durch zwei Wrapper-Klassen, mit denen die Interfaces jeweils gemappt werden.

Ein Mapping ist auch für OLE-Automation vorgesehen.

Tools zur Erstellung von OpenDoc Parts

Um die Erstellung von OpenDoc-Parts zu vereinfachen, muß es selbstverständlich Tools geben. Diese Tools sind im Moment sowohl von Apple wie IBM in der Entwicklung. Sehr populär in der Apple-Welt ist das »OpenDoc Development Framework« (ODF). Von Taligent (jetzt 100% IBM) entwickelt wurde das »Compound Document Framework«, das noch in diesem Jahr für OS/2 mit OpenDoc-Unterstützung ausgeliefert werden soll.

Als Teil des Toolkits selber kommt ein Wizzardprogramm, der sogenannte »Partmeister«.

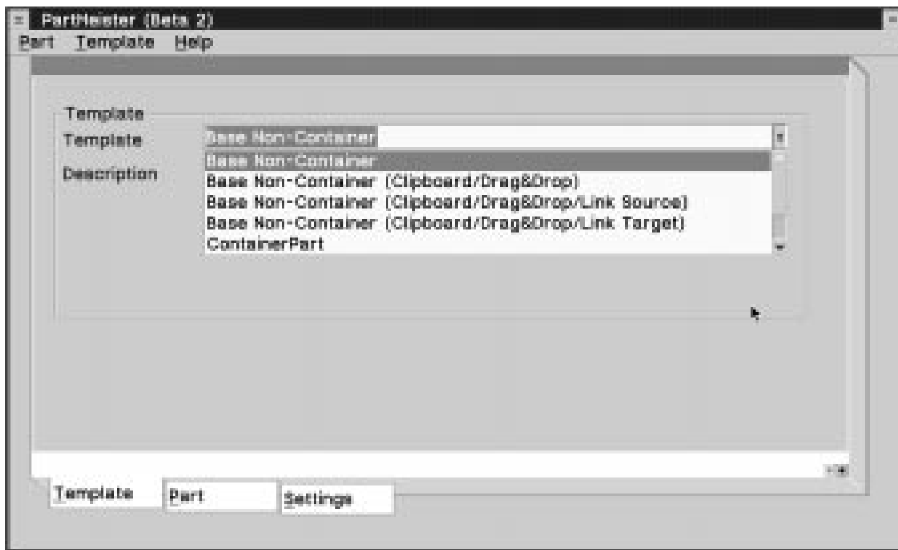


Abb. 7.21: Partmeister-Templates

Von diesen Schablonen, die alle von *ODPart* abgeleitet sind, kann eine gewählt werden, deren Optionen angepaßt sind und mit Attributen wie Namen, Ersteller usw. versehen werden. Das Tool »Partmeister« generiert dann daraus drei Dateien

– eine IDL-Datei

– ein CPP-Datei

– eine Make-Datei, die für den ausgewählten Compiler angepaßt ist.

Der folgende IDL-Code wurde von Partmeister generiert, nachdem wir spezifiziert haben, daß wir für eine Grafik einen Basis-Part generiert haben wollen, der bereits Drag&Drop unterstützt.

Die generierte IDL-Datei definiert eine Klasse *CntrDrDropLink*, die von der Klasse *IBM_BaseNonContainer* abgeleitet ist, die wiederum von ODPart erbt.

Im Override-Teil wird angegeben, welche Methoden dabei überschrieben werden müssen. Für diese Methoden legt das Partmeister-Tool dann bereits Methodenrümpfe in der CPP-Datei an.

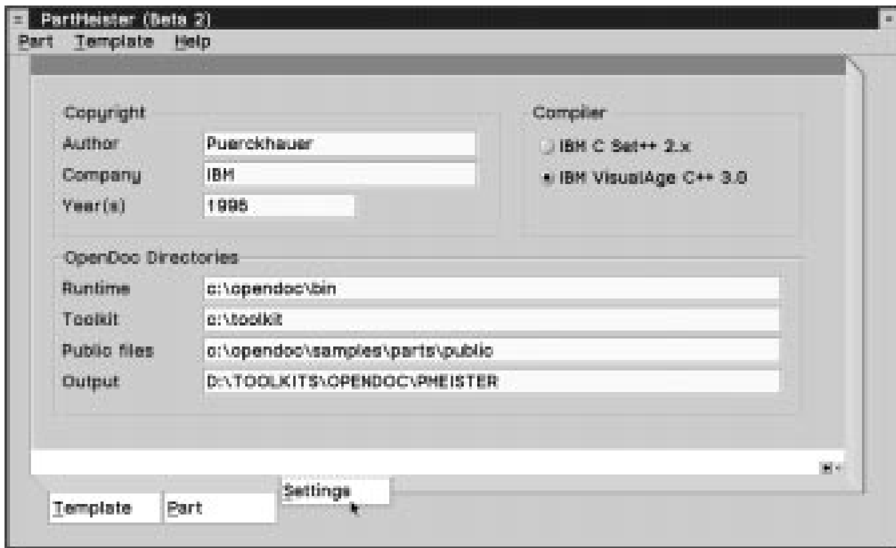


Abb. 7.22: OpenDoc Partmeister

Als Beispiel hier eine IDL-Datei, die vom Partmeister-Tool generiert wurde.

```
#ifdef __PRIVATE__
//#
//#    File:          CONTI.IDL
//#
//#    Description:   IDL for CntrDrDropLink
//#
//#    Written by:    Puerckhauer
//#
//#    Generated by:  IBM PartMeister Beta 2
//#
#endif
//#    Copyright:     (c) 1996 by IBM
//#                   – all rights reserved

#ifndef _CONTI_
#define _CONTI_

#ifndef _BASENC_
#include <basenc.idl>
#endif

// meatless
interface M_CntrDrDropLink;


interface CntrDrDropLink : IBM_BaseNonContainer
 {
    attribute string text;

    #ifdef __SOMIDL__
    implementation
     {
        metaclass      = M_CntrDrDropLink;
        majorversion   = 1;
        minorversion   = 0;
        dllname        = "Conti.dll";

        releaseorder:
            _set_text,
            _get_text;
        override:
            somDefaultInit,
            somDestruct,
            InitPart,
            InitPartFromStorage,
            Draw,
            Externalize,
            CloneInto,
```

```
            MenuCopyValid,
            MenuPasteValid,
            PopulateClipboardStorageUnit,
            ReceiveClipboardStorageUnit,
            DragValid,
            DropValid,
            PopulateDragStorageUnit,
            ReceiveDropStorageUnit,
            LinkSourceValid,
            PopulateLinkStorageUnit;

        // method modifiers
        somDefaultInit: init;
     };
    #endif
 };

interface M_CntrDrDropLink : M_IBM_BaseNonContainer
 {
    #ifdef __SOMIDL__
    implementation
     {
        override:
            clsGetODPartHandlerName,
            clsGetODPartHandlerDisplayName,
            clsGetODPartKinds,
            clsGetOLE2ClassId,
            clsGetWindowsIconFileName;
     };
    #endif
 };
#endif
```

Dieses Code-Gerüst muß dann vom Entwickler noch mit der Part-spezifischen Logik angereichert werden:

```
SOM_Scope void  SOMLINK InitPart(CntrDrDropLink *somSelf,  Environment *ev,
                                 ODStorageUnit* storageUnit,
                                 ODPart *partWrapper)
{
    CntrDrDropLinkData *somThis = CntrDrDropLinkGetData(somSelf);
    CntrDrDropLinkMethodDebug(kPartHandlerName,"InitPart");

    CntrDrDropLink_parent_IBM_BaseNonContainer_InitPart(somSelf, ev,
                                                   storageUnit, partWrapper);
    CommonInitPart (somSelf, ev);

    // store kind to storage unit
    storageUnit->AddProperty (ev, kODPropContents)->AddValue (ev, kKind);
```

```
    // set preferred editor property to this part
    storageUnit->AddProperty (ev, kODPropPreferredKind)->AddValue (
                                                               ev, kODISOStr);
    StorageUnitSetValue (storageUnit, ev, strlen (kKind) + 1, kKind);

    _text = (string) SOMMalloc (strlen ("CntrDrDropLink") + 1);
    strcpy (_text, "CntrDrDropLink");
}

SOM_Scope void  SOMLINK InitPartFromStorage(CntrDrDropLink *somSelf,
                                            Environment *ev,
                                            ODStorageUnit* storageUnit,
                                            ODPart *partWrapper)
{
    CntrDrDropLinkData *somThis = CntrDrDropLinkGetData(somSelf);
    CntrDrDropLinkMethodDebug(kPartHandlerName,"InitPartFromStorage");

    CntrDrDropLink_parent_IBM_BaseNonContainer_InitPartFromStorage(
                                 somSelf, ev, storageUnit, partWrapper);
    CommonInitPart (somSelf, ev);

    // retrieve part data from persistent storage
    storageUnit->Focus (ev, "Text", kODPosUndefined, kODKindPlainText, 0,
                                                 kODPosUndefined);
    long size = storageUnit->GetSize (ev);
    if (_text != NULL)
        SOMFree (_text);
    _text = (string) SOMMalloc (size);
    StorageUnitGetValue (storageUnit, ev, size, _text);
}
SOM_Scope void  SOMLINK Externalize(CntrDrDropLink *somSelf,  Environment *ev)
{
    CntrDrDropLinkData *somThis = CntrDrDropLinkGetData(somSelf);
    CntrDrDropLinkMethodDebug(kPartHandlerName,"Externalize");

    ODStorageUnit* su = somSelf->GetStorageUnit (ev);

    if (su->Exists (ev, kODPropContents, kKind, 0))
     {
        su->Focus (ev, kODPropContents, kODPosUndefined, kKind, 0,
                                        kODPosUndefined);
        su->Remove (ev);
        su->AddValue (ev, kKind);
     }
    else
        su->AddProperty (ev, kODPropContents)->AddValue (ev, kKind);

    // store persistent data
    su->AddProperty (ev, "Text")->AddValue (ev, kODKindPlainText);
```

```
    StorageUnitSetValue (su, ev, strlen (_text) + 1, _text);

    CntrDrDropLink_parent_IBM_BaseNonContainer_Externalize(somSelf, ev);
}

SOM_Scope void  SOMLINK CloneInto(CntrDrDropLink *somSelf,
                                  Environment *ev, ODDraftKey key,
                                  ODStorageUnit *toSU, ODFrame *scopeFrame)
{
    CntrDrDropLinkData *somThis = CntrDrDropLinkGetData(somSelf);
    CntrDrDropLinkMethodDebug(kPartHandlerName,"CloneInto");

    ODStorageUnit *su    = somSelf->GetStorageUnit (ev);
    ODDraft       *draft = su->GetDraft (ev);

    somSelf->Externalize (ev);

    ODID scopeFrameID = 0;
    if (scopeFrame != kODNULL)
        scopeFrameID = scopeFrame->GetStorageUnit (ev)->GetID (ev);
    su->CloneInto (ev, key, toSU, scopeFrameID);
}
```

7.3.17 Das Compound Document Framework

Das »Compound Document Framework« ist ein sehr mächtiges und einfach zu programmierendes Framework, das die Unterschiede zwischen Komponentarchitekturen wie OLE und OpenDoc abstrahiert, so daß aus diesem Framework sowohl OLE- wie OpenDoc-Parts generiert werden können.

Das Compound Document Framework besteht aus vier Basis-Klassen, einer Model-Klasse, einer View-Klasse, einer Selecion-Klasse und einer Command-Klasse, die durch einfache Spezialisierung an eigene Anforderungen angepaßt werden kann.

Hier definiert man die Daten, mit denen man arbeiten möchte, in der Model-Klasse, definiert seine Benutzerschnittstelle in der View-Klasse, und definiert, welche Selektionen zulässig sind und welche Commands darauf ausgeführt werden dürfen. Alle anderen Funktionen wie Drag&Drop, Embedding, Linking, Scriptunterstützung und Interoperabilität werden vom Framework selbst unterstützt.

Die Views sind strikt vom Modell getrennt. Für ein Modell können mehrere Views gleichzeitig bestehen Die »Invoice«-, »Accounts receivable«- und »Rolodex«-Anwendungen benützen unterschiedliche Views desselben Modells »Account Info«.

»Account Info« hat in diesem Beispiel drei Attribute:

»Name/Address«, »Phone No.« und »Credit Info«.

Wenn die Daten im Modell, z.B. die Adresse, sich ändern, werden alle Views des Modells benachrichtigt, so daß sie sich erneuern können. Die Presenter-Klasse bearbeitet alle Benutzer-Aktionen.

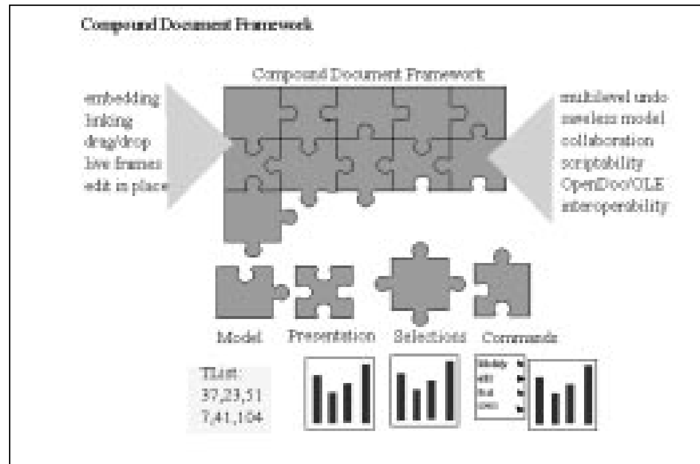


Abb. 7.23: Compound Document Framework

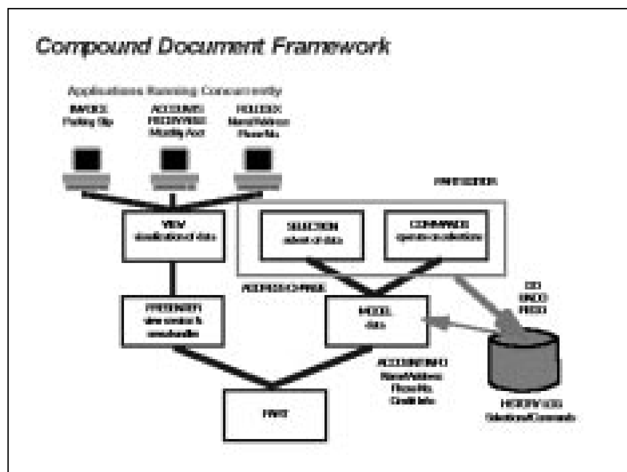


Abb. 7.24: Das Zusammenspiel im Framework

In einem View kann der Benutzer eine Selektion treffen, mit der der ein Datenbereich des Modells spezifiziert wird, wie z.B. die Adresse.

Eine Selektion im Compound Document Framework ist völlig unabhängig von der Art, in der sie erstellt wird. Sie kann über eine Maus-Aktion, Tastatureingabe, Aktionsleiste oder ein Script definiert werden.

Diese Selektionen können für die Manipulation der Daten verwendet werden. Eine Selektion ist mit einem Set von Commands assoziiert, die spezifisch für diese Selektion sind, um die Daten zu verändern.

Auch diese Commands können auf unterschiedliche Art und Weise gestartet werden, nämlich durch die Aktionsleiste, Kommandozeile, eine Mausnachricht oder ein Script. Commands agieren immer auf Selektionen, nie direkt auf den Daten des Modells. Die Aktionen, die als Command-Selection-Paare ausgeführt werden, speichert das CDF in einer History-Log-Datei, die dadurch ein Multilevel-Undo erlaubt, das bereits im Framework implementiert ist.

Dadurch, daß die Command-Selection-Paare unabhängig in der Art der Erstellung sind, ist die Scriptunterstützung »for free«.

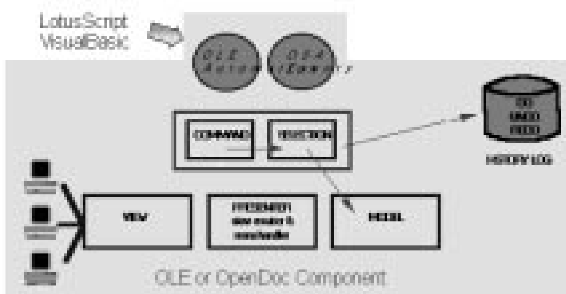


Abb. 7.25: OLE versus OpenDoc

Eine abgespeckte Version des CDF ist bereits in der Visual-Age-C++-Version für Windows enthalten, die im April dieses Jahres auf den Markt kam, allerdings ohne Command- und Selection-Klassen und ohne OpenDoc-Unterstützung. Eine vollständige Version soll aber noch dieses Jahr zur Verfügung stehen.

7.3.18 Ausblick

OpenDoc bietet eine technisch ausgereifte Architektur, die in vielen Aspekten anderen Architekturen überlegen ist. Ein Problem wird durch heutige Komponentenarchitekturen nur unzulänglich gelöst: das Problem der Binärportabilität von Komponenten. Zwar sind die Schnittstellen von OpenDoc plattformunabhängig, die Komponenten selber müssen aber für jede Plattform neu umgewandelt werden, auch wenn man portable Klassenbibliotheken wie IBM-OpenClass verwendet. Dieses Problem hat sich durch den Boom des WWW dramatisch geändert. So geht die Entwicklung immer mehr dahin, WWW-Browser wie z.B. den von Netscape, mit Container Funktionalität über Komponenten auszurüsten. Java-Applets sind binär portabel, da sie auf einer virtuellen Maschine laufen, die entweder im Web-Browser oder im Betriebssystem wie bei OS/2 Warp Version 4 implementiert ist.

Löst HTML mit Java-Applets also Komponentenarchitekturen wie OLE und OpenDoc ab? Sicher nicht! Beides sind mehr oder weniger komplementäre Technologien. Alle Schnittstellen, die für die Kooperation und Interaktion von verschiedenen Komponenten notwendig sind, wie gemeinsames Abspeichern unterschiedlicher Datenformate, Drag&Drop von verschiede-

nen Komponenten, Linking und so weiter sind durch Java und HTML überhaupt nicht adressiert. IBM hat angekündigt, die OpenDoc-Technologie in enger Kooperaion mit SUN mit Java zu verschmelzen. Das Codewort für diese Entwicklung ist Arabica. Erste Prototypen wurden zum Beispiel auf der IBM-Object-Konferenz in San Francisco im Juni 1996 vorgestellt. Auch Netscape arbeitet intensiv an einer OpenDoc-Integration, so daß die Zukunft wohl aus interoperablen WWW-Parts und -Containern besteht, die von und zu WWW-Browsern aus Verbunddokumenten mit entfernten Links zum Beispiel über WWW-DB2-Gateways zu Daten auf einer Host-Datenbank oder mit Links in Notes-Datenbanken beliebig ausgetauscht werden können .

7.4 DIVE

von Wolfgang Engel

Die Direct Interface Video Extension – kurz DIVE-API – dient dazu, Unterhaltungs-, Lern- und Spielesoftware eine möglichst schnelle Darstellung ihrer Bilder auf dem Bildschirm zu ermöglichen. Bevor die DIVE-API mit OS/2 Warp V3 eingeführt wurde, war dies nur auf zwei Wegen möglich: Entweder durch die Verwendung einer Vollbild-OS/2-VIO-Session oder des Presentation Manager.

Beide Lösungen hatten zahlreiche Nachteile. Mit der VIO-API ist zwar eine schnelle Grafikdarstellung möglich, jedoch ist sie nach wie vor 16-bittig und erfordert somit eine Umsetzung (thunking), wenn sie aus 32-Bit-Anwendungen verwendet werden soll. Außerdem ist eine synchronisierte Hardware- und Dateiformat-unabhängige Multimedia-Unterstützung mit dem Multimedia Presentation Manager (MMPM/2) nicht möglich.

Auf der anderen Seite könnte eine Presentation-Manager-Anwendung, die sich des Graphics-Programming-Interface-APIs (GPI) bedient, zwar den MMPM/2 in Anspruch nehmen, jedoch wäre damit ein direkter Zugriff auf die Grafikkarte zur Wahrung der Integrität der Arbeitsoberfläche nicht möglich. Die Darstellungsgeschwindigkeit einer solchen Presentation-Manager-Anwendung wäre deshalb, verglichen mit einer DIVE-Anwendung, sehr viel langsamer.

7.4.1 Was kann DIVE?

DIVE nutzt zur Darstellung die Möglichkeiten der neuen Grafikkartengeneration, mit der die Farbkonvertierung, Bildvergrößerung und -verkleinerung und das sogenannte »Clipping« größtenteils von der Grafikkarte und vom Prozessor übernommen wird. Durch die schnelle Farbkonvertierung eignet sich DIVE nicht nur gut zur Spieleprogrammierung, sondern auch zur Darstellung von Videofilmen unter OS/2, bei denen unterschiedliche Farbformate in das jeweilige OS/2-Format möglichst schnell konvertiert werden müssen.

Das DIVE-API ermöglicht es dem Programmierer, auf zwei unterschiedliche Arten auf die Grafikkarte zuzugreifen. Für jede der beiden Zugriffsarten stellt es eine eigene Schnittstelle zur Verfügung. So gibt es – wie unter DOS – die Möglichkeit, die Anfangsadresse des Grafikkartenspeichers zu ermitteln, um dann in diesen die Bildinformationen direkt zu schreiben. Diese Schnittstelle soll hier Direct-Access-API oder Low-Level-DIVE-API genannt werden. Auf der anderen Seite kann der Programmierer aber auch den IBM-DIVE-»Rundum-Service« wählen. Er sollte hierzu die sogenannte DIVE-Blitter-API verwenden, die den Zugriff auf die Grafikkarte vollständig in einfache Funktionsaufrufe, die jegliche hardwarenahe Programmierung überflüssig machen, verkapselt. Eine DIVE-Blitter-Anwendung verwaltet dabei die Grafikkarte und den Grafikkartenspeicher ohne Zutun des Programmierers selbständig, auch im Verhältnis zu anderen Programmen. Der DIVE-Blitter übernimmt im einzelnen folgende Aufgaben:

– Scaling des OS/2-Fensters: Der Benutzer kann das Fenster einer mit DIVE programmierten Anwendung, wie jedes andere Presentation-Manager-Fenster, stufenlos in der Größe verändern. Die dabei notwendig werdende Anpassung des Bildes an den Rahmen des Fensters übernimmt DIVE.

– Remapping der Farben: Der DIVE-Blitter kann die Anpassung der von der Anwendung verwendeten Farben an die von der Arbeitsoberfläche verwendeten Farben vornehmen. Dabei kann es nicht nur vorkommen, daß ein Programm mit 256 Farben in einem Video-Modus mit 65535 Farben betrieben wird, sondern es ist auch möglich, daß die Arbeitsoberfläche zum Beispiel im 256-Farben-Modus eine Farbe verwendet, die im Programm nicht vorkommt.

– Clipping der Ausgabefläche: Wenn andere Fenster sich mit dem Ausgabefenster des DIVE-Blitters überschneiden, sollte dessen Ausgabe nur noch in die nicht verdeckte Ausgabefläche erfolgen.

- In Verbindung mit dem Game-Server: Umschaltung zwischen Fenster und Full-Screen oder Vollbild-Modus. Der Vollbild-Modus erlaubt eine Auflösung von 320 x 200 x 8 Pixel.

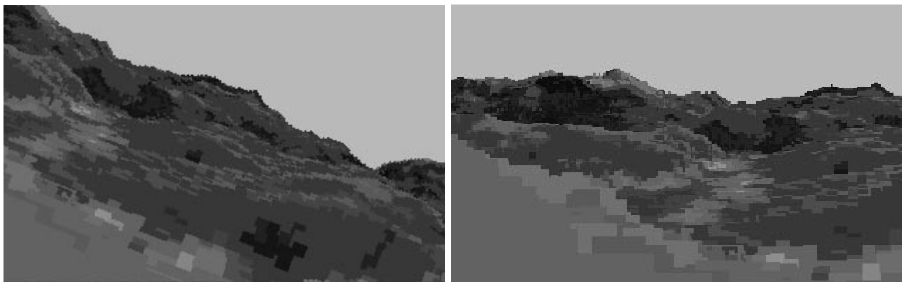


Abb. 7.26: Bild einer Helicopter-Simulation, die mit DIVE geschrieben wurde. Wir bewegen uns mit 320 km/h zwei Meter über dem Erdboden (zu finden im Compuserve Forum OS2DF1 MULTIMEDIA unter werwolf.zip und an vielen anderen Stellen)

Wie funktioniert nun DIVE? DIVE besteht im wesentlichen aus einer Dynamic Link Library (DLL). Diese DLL benötigt außerdem den Treiber ssmdd.sys aus dem MMPM/2.



Abb. 7.27: Das Zusammenspiel von DIVE mit der Grafikkarte

Ein typisches DIVE-Programm läuft als Presentation-Manager-Anwendung, wobei ein eigener Thread für die DIVE-Anzeige-Funktionen bereitgestellt werden muß. Dieser DIVE-Client oder das Programm, das sich der DIVE-API bedient, ruft die einzelnen Funktionen der DIVE.DLL auf. Die dort befindliche Display-Engine prüft dann, ob die gewünschten Hardwarefunktionen von der Grafikkarte unterstützt werden und emuliert diese, falls dies nicht der Fall sein sollte. Der eigentliche Zugriff auf die Grafikkarte erfolgt über Erweiterungen des Grafikkartentreibers. Ein Aufruf von GreEscape mit DEVESC_VRAMALLOC alloziert beispielsweise Grafikkartenspeicher (eine ausführliche Auflistung findet sich in der »Presentation Device Driver Reference for OS/2« unter dem Kapitel: »Presentation Manager Function Categories«).

Gegenüber DIVE verfügt das sogenannte Enhanced DIVE (EnDIVE) über wesentlich erweiterte Möglichkeiten der Interaktion mit der Grafikkarte. Dabei wurde vor allem die Unterstützung der Wiedergabe von Videoaufnahmen stark erweitert. EnDIVE ist vollständig in die DIVE.DLL integriert. Die erweiterte Funktionalität wird bei Verfügbarkeit automatisch genutzt, so daß sich der Programmierer nicht darum zu kümmern braucht.

DIVE sollte mit jeder Grafikkarte, die über OS/2-Treiber verfügt, funktionieren. Demgegenüber verlangt EnDIVE einen eigens dafür geschriebenen Treiber, über den zur Zeit der Drucklegung des Buchs nur ELSA- und Matrox-Grafikkarten verfügen.

DIVE kann folgende Formate aufnehmen:

– CLUT 8 (256 Farben) – LUT8

– 8 – bit greyscale – GREY

– RGB 16 (5-6-5, 5-5-5) – R565, R555, R664

– RGB 24 (R-G-B, B-G-R) – RGB3, BGR3

– RGB 32 (R-G-B, B-G-R) – RGB4, BGR4

– YUV 9-DVI/Indeo three-plane color subsampled – YUV9

– YUV 422 – Y422

– YUV CCIR601 – three-plane 2x2 color subsampled (MJPEG, MPEG) – Y2X2

– YUV CCIR601 – three-plane 4x4 color subsampled – Y4X4

Die Ausgabeformate sind:

– CLUT 8 (256 Farben)

– RGB 16 (5-6-5, 5-5-5, 6-6-4)

– RGB 24 (R-G-B, B-G-R)

– RGB 32 (R-G-B-x, B-G-R-x)

– YUV 422 (Y-U-Y-V)

7.4.2 Der Game-Server als Erweiterung von DIVE

Soll eine DIVE Anwendung auch in einem speziellen Vollbildschirmmodus betrieben werden, benötigt diese Anwendung auch noch die GAMESRVR.DLL. Mit dem darin enthaltenen Game-Server kann einer DIVE-Anwendung ein Vollbildschirmmodus zur Verfügung gestellt werden, der eine Mischung aus einer OS/2-VIO-Vollbildsitzung und einer Presentation-Manager-Sitzung darstellt. Der Game-Server öffnet eine VIO-Vollbildsitzung und blendet die Bildinformationen des DIVE-Presentation-Manager-Programms dort in den Grafikspeicher.

Dabei soll hier aber nicht verschwiegen werden, daß der Vorteil, mit dem Game-Server über einen Vollbildmodus verfügen zu können, mit erheblichen Beschränkungen erkauft werden muß:

So ist eine DIVE-Anwendung, die sich des Game-Servers bedient, auf eine Auflösung von 320x200 bei 256 Farben beschränkt. Innerhalb des Vollbildmodus kann natürlich nicht die GPI-API benutzt werden, so daß die von OS/2 gewohnten Dialoge, Menüs oder Schriften nicht verwendet werden können. Zudem blendet der Game-Server den Mauscursor vollständig aus.

Freilich werden diese Beschränkungen für Spiele keine große Rolle spielen.

7.4.3 DIVE als Portierungs-API für DOS-Spiele

Gerade die Benutzung des DIVE-Blitters macht die Portierung von DOS-Programmen, was die hardwarenahe Programmierung anbelangt, einfach. DIVE eignet sich deshalb hervorragend zur kostengünstigen Portierung von DOS-Spielen nach OS/2.

Viele DOS-Spiele verfügen über eine der DIVE-API ähnliche Architektur, was natürlich von den Entwicklern von DIVE beabsichtigt war. Der Speicherbereich des DIVE-Blitters, in den die Anwendung ihre Bildinformationen schreibt, ist durchgehend. Auf ihn kann daher wie zum Beispiel im Grafikkartenmodus 13h in jeder Darstellungsgröße am Stück zugegriffen werden. Zudem unterstützt DIVE in Verbindung mit dem Game-Server auch den beliebten 256-Farben-Modus im Vollbildmodus, der nach wie vor von den meisten Spielen genutzt wird.

Meist stellt sich bei einer Portierung nach OS/2 die Frage, inwiefern vorhandener Assemblerquelltext ohne Veränderung weiter verwendet werden kann. Viele DOS-Spiele benutzen in ihrer innersten Schleife self-modifying code. Diese Technik ist wegen dem unter OS/2 schreibgeschützten Codeobjekt nicht ohne weiteres anwendbar. Allerdings – und das soll hier nur angeschnitten werden – kann das jeweilige Teilstück, in dem sich self-modifying Code befindet, auch in ein Datenobjekt eingelesen werden, so daß, wenn der ursprüngliche Quellcode weitgehend unverändert bleiben soll, auch self-modifying Code keine Portierungshürde darstellen sollte.

7.4.4 Die Grenzen von DIVE

DIVE kann fortlaufend immer nur in diesselbe rechteckige Ausgabefläche ausgeben. Bei jeder Änderung der Ausgabefläche wird zunächst die Ausgabefunktion von DIVE angehalten. Deshalb könnte DIVE zum Beispiel nicht dazu verwendet werden, ein Sprite möglichst schnell über den Bildschirm zu bewegen. Andererseits eignet es sich gerade aufgrund dieses Umstandes gut für die Videowiedergabe und sogenannte »first-person perspective games«, wie zum Beispiel D**m oder Qu***, in denen die Ausgabefläche den ganzen Bereich des Ausgabefensters unverändert einnimmt.

Natürlich wäre es auch möglich, in einem Ausgabefenster zum Beispiel fünf Rechtecke als Ausgabefläche von fünf verschiedenen DIVE-Instanzen anzugeben. Beispielsweise um ein Programm für Präsentationen zu schreiben. Allerdings sollte dabei berücksichtigt werden, daß jede DIVE-Instanz eine erhebliche Menge Speicher für die Grafik und die Verwaltung benötigt, der dann im Computerzielsystem in Form von Hauptspeicher vorhanden sein müßte. DIVE ist auch kein Ersatz für die GPI, da es keine Funktionen gibt, um Grafikprimitive wie Linien, Rechtecke usw. anzuzeigen. Wie unter DOS muß der Programmierer entweder eigene Routinen für diese Aufgaben schreiben oder auf hierfür spezialisierte 3D-Bibliotheken (z.B. BRender oder OpenGL für OS/2) zurückgreifen.

7.5 Open32 und OS/2-Portabilität

von Michael Frieß

Vielleicht ist Ihnen in der OS/2-Fachpresse schon einmal der Begriff *Open 32* aufgefallen. Während der Installation von OS/2 Warp 4.0 erscheint eine kurze Information über die neu integrierten Schnittstellen OpenDoc, OpenGL und Open32. Im Gegensatz zu den zwei anderen genannten Schnittstellen erhalten Sie über Open32 keinen weiteren Hinweis. Es wird automatisch mitinstalliert. Falls Sie sich in der Online-Dokumentation auf die Suche machen, werden Sie weder im Hilfeindex noch im Glossar fündig. Ein kleiner Absatz im Überblick zum OS/2-Warp-System informiert Sie, daß dies die neue Bezeichnung für die *Developer API Extensions* (DAPIE) ist und zahlreiche Programme in der Entwicklung sind, die diese Schnittstelle benützen. Die erwähnten Programme »Lotus WordPro for OS/2« und »PhotoSuite« befinden sich auf der Application-Sampler-CD als Beta- beziehungsweise Demoversion.

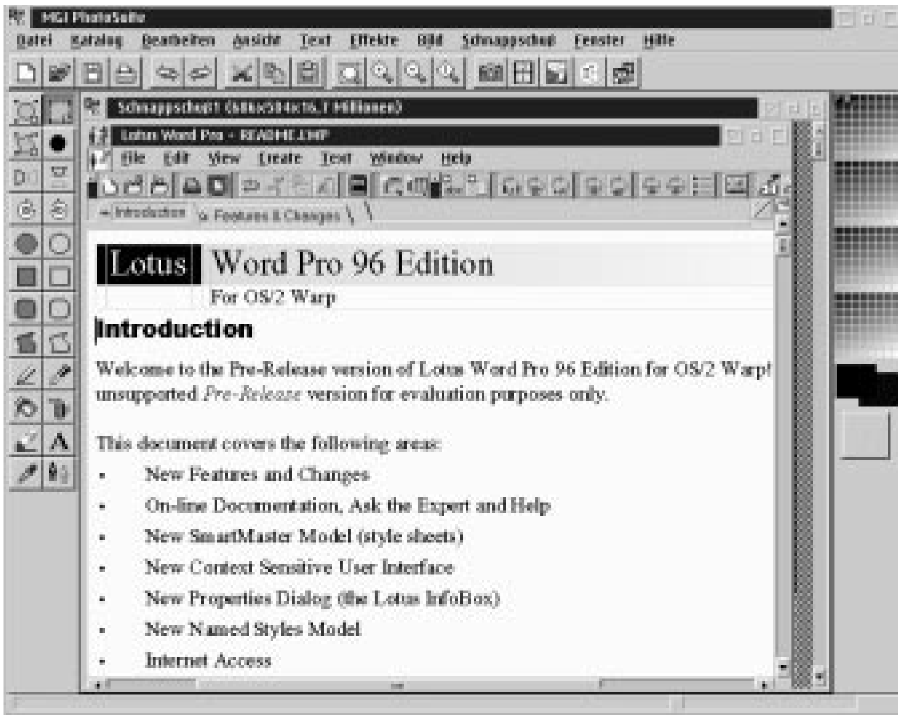


Abb. 7.28: Die Open32-Anwendungen MGI PhotoSuite und Lotus WordPro

Sie können diese OS/2-Anwendungen installieren, ohne daß Sie oder ohne daß das Installationsprogramm überprüfen muß, ob die Open32-Schnittstelle installiert ist. Als Benutzer brauchen Sie nichts über Open32 zu wissen, die entsprechenden Anwendungen verhalten sich genauso wie bisherige OS/2-Anwendungen. Die Open32-Schnittstelle steht auch für OS/2 Warp Version 3 ab FixPak 17 zur Verfügung. Besitzer der alten OS/2-Warp-Version kommen somit gleichfalls in den Genuß, diese Anwendungen benutzen zu können.

Doch welchen Nutzen bringt diese neue Schnittstelle?

Open32 stellt eine definierte Teilmenge des Win32-API unter OS/2 Warp bereit. Es bietet damit Entwicklern die Möglichkeit, mit einer einzigen Codebasis gleichzeitig OS/2 Warp und die diversen Windows-Versionen abzudecken. Damit können Software-Hersteller ihre reinen Windows-Anwendungen mit vergleichbar geringem Aufwand auch auf OS/2 Warp anbieten und somit weitere potentielle Käufer erreichen.

Ihnen als OS/2-Anwender nützt also Open32 indirekt durch ein größeres Angebot an echten OS/2-Anwendungen. Diese Anwendungen sind nicht nur echte 32-Bit-Anwendungen, sondern können auch zusätzliche OS/2-Funktionen benützen und haben dadurch zahlreiche technische Vorteile gegenüber entsprechenden 16-Bit-Windows-Anwendungen unter Win-OS/2. Vor allem haben Sie aber als Endanwender auch die Gewährleistung, daß der jeweilige Anbieter die Anwendung unter OS/2 unterstützt, während es Ihnen bei einer Anwendung unter Win-OS/2 passieren kann, daß Sie vom Hersteller keine Unterstützung erhalten und nur auf IBM weiterverwiesen werden.

Der restliche Teil des Kapitels richtet sich vorrangig an Entwickler oder interessierte Leser. Es erklärt zuerst grundsätzliche Aspekte der portablen Anwendungsentwicklung und bietet einen historischen Abriß der OS/2-Portabilität. Dies ermöglicht es Ihnen, Open32 im geschichtlichen und derzeitigen Umfeld richtig einzuordnen. Die weiteren Abschnitte bieten dann technische Informationen über Open32 und die Vorgehensweise, wie Sie als Entwickler die Schnittstelle benützen können.

7.5.1 Portable Anwendungsentwicklung

Portabilität ist ein wesentliches Qualitätsmerkmal von Software. Es bietet dem Anwender Investitionsschutz. Ist eine Software über mehrere Plattformen – üblicherweise Betriebssysteme oder Hardware – portabel, können die mit der Software gewonnenen Fähigkeiten und erstellten Informationen nach einem Umstieg auf eine andere Plattform weiterhin benützt werden.

Grundsätzlich läßt sich Portabilität entweder während der Laufzeit oder während der Entwicklung einer Software verwirklichen.

– Binäre Portabilität

Ausführbare Programmdateien (».exe« oder ».dll«) und/oder Datendateien einer Software sind über Plattformen hinweg lauffähig und benützbar.

– Portabilität des Quelltexts

Vom Entwickler erstellte oder generierte Quelldateien einer Software können auf mehreren Plattformen kompiliert und gebunden werden. Die Software kann dann auf diesen Plattformen ausgeführt werden.

Ausführbare Programmdateien im üblichen Sinne enthalten die Programmlogik in Form der speziell für einen Prozessor erzeugten Maschinenbefehle. Die binäre Portabilität auf einer Maschine mit einem anderen Prozessor kann also nur durch ein Softwareprogramm, den »Emulator«, geschehen, der das Verhalten des ursprünglichen Prozessors simuliert und entsprechend umsetzt. Der Einsatz eines Emulators bringt jedoch meist drastische Einbußen in der Performance mit sich.

Zusätzlich hängen die Programmdateien auch von dem Vorhandensein der benutzten Funktionen des Betriebssystems ab, für das sie erzeugt worden sind. Existiert also ein zweites Betriebssystem auf der gleichen Hardwareplattform, könnte es prinzipiell bestehende Anwendungen ohne Emulator ausführen, sofern es die Schnittstellen des originären Betriebssystems abbildet.

Ein Betriebssystem oder ein Emulator erbringt also die Leistung der binären Portabilität während der Laufzeit. Beim zweiten Ansatz liegt es in Ihrer Verantwortung als Entwickler, für Portabilität des Quellcodes in der Entwicklungsphase zu sorgen. Sie beeinflussen die Portabilität maßgeblich durch die Auswahl der Entwicklungskomponenten. Als wesentliche Entscheidungskriterien können Sie offene Standards und den Verbreitungsgrad der Komponenten heranführen.

Der Begriff des »offenen Standards« wird von Softwarefirmen unterschiedlich interpretiert. Im folgenden ist damit ein Standard gemeint, der durch ein Gremium erarbeitet und gepflegt wird, in dem alle Betroffenen die Möglichkeit haben, aktiv mitzuwirken. Einzelne Firmen oder geschlossene Gremien bezeichnen aber auch Schnittstellen als offen, wenn sie deren Dokumentation offengelegt haben. Durchaus können solche Schnittstellen einen solchen Verbreitungsgrad erreichen, daß sie als De-facto-Standard angesehen werden. Falls diese vom Hersteller oder von Drittanbieter auf mehreren Plattformen verfügbar sind, können sie als Basis für portablen Code verwendet werden. In diesem Fall haben Sie jedoch als Entwickler keinen Einfluß auf weitere Versionen der Schnittstelle und daß diese auch zukünftig auf allen Plattformen gepflegt werden. Sie gehen somit eine langfristige Abhängigkeit zu dem jeweiligen Hersteller ein.

Sie sehen in der folgende Tabelle Entwicklungskomponenten, die vorrangig die Portabilität bestimmen, und entsprechende Standards oder verbreitete Beispiele.

Element	**Standard**	**Verbreitet**
Programmiersprache	ANSI C, C++ Draft	Pascal, xBase, Java
	ANSI Cobol	–
Compiler/Interpreter	–	Turbo Pascal, Visual Basic
API	OpenGL, SQL, TCP/IP	ODBC
	Motif !–	
Dateiformat	MPEG, JPEG	GIF, ZIP, DBF
Klassenbibliothek	STL, Collection Classes	AWT
Objektmodell	CORBA	COM
Komponentenarchitektur	OpenDoc	OLE

In der traditionellen prozeduralen Softwareentwicklung waren Programmiersprache, Compiler/Interpreter, APIs und Dateiformate für die Portabilität entscheidend. Durch die ständig wachsende Verbreitung der Objektorientierung werden Klassenbibliotheken, das Objektmodell und die Komponentenarchitektur die Portabilität maßgeblich beeinflussen.

Da bekanntlich die Entwicklung der grafischen Benutzerschnittstelle einen großen Teil am Gesamtaufwand der Softwareerstellung darstellt und gerade diese sich stark von Betriebssystem zu Betriebssystem unterscheidet, bieten zahlreiche Hersteller hierfür portable Bibliotheken (siehe [1]) an. Diese sind meistens als C-Funktionsbibliothek oder als C++-Klassenbibliothek realisiert. Sie decken in der Regel eine große Anzahl von Plattformen ab, bieten aber entsprechend eher nur den gemeinsamen Nenner an Funktionalität an.

Wie aus einem technischen Papier der IBM [2] hervorgeht, ist es ihr Ziel, Entwickler bei der Softwareerstellung auf Basis einer Codebasis für mehrere Plattformen zu unterstützen. Hierzu bietet sie eine Reihe von Werkzeugen, Technologien und Systemkomponenten an:

– VisualAge-Familie

Eine Reihe von Entwicklungsprodukten für unterschiedliche Programmiersprachen und mehrere Plattformen, die jeweils Werkzeuge für die visuelle Programmerstellung enthalten.

– IBM Open Class-Bibliothek

Die C++-Klassenbibliothek ist zusammen mit den C++-Produkten auf den verschiedenen Plattformen erhältlich. Sie enthält Klassen für die Benutzerschnittstelle, den Datenbankzugriff, Mengen von Objekten und grundlegende Basisklassen.

– OpenDoc

Die Komponentenarchitektur ist auf mehreren Plattformen verfügbar. Sie wurde ursprünglich von IBM, Apple und Novell gemeinsam entwickelt und ist nun von der »Object Management Group« (OMG) in den Standard aufgenommen worden.

– Open32

Mit Open32 wird eine Untermenge der 32-Bit-Windows-APIs auch unter OS/2 Warp angeboten.

– Source Migration Analysis Reporting Toolset (SMART)

Ein Werkzeugkasten, der die Portierung von Windows- und 16-Bit-OS/2-Code auf 32-Bit-OS/2 beträchtlich erleichtert.

– Hyperwise

Ein WYSIWIG-Editor zur Erstellung von Hilfetexten und Online-Dokumenten für OS/2 und Windows.

Die folgenden Betrachtungen beschränken sich auf die Portabilität speziell zwischen OS/2 und Windows. Dies hat zwei Gründe. Zum einen laufen beide Betriebssysteme überwiegend auf der gleichen Hardware, was die binäre Portabilität mit Win-OS/2 überhaupt ermöglicht. Zum anderen beruht Open32 gerade auf der engen Verwandtschaft dieser Betriebssysteme, bedingt durch deren gemeinsame Geschichte.

7.5.2 Drei Betriebssysteme, eine Hardware

DOS, OS/2 und Windows sind verschiedene Betriebssysteme, die auf der gleichen Hardwareplattform, den IBM-kompatiblen Rechnern verfügbar sind. Die Programmierschnittstelle von DOS ist sehr hardwarenah gelöst. Anwendungsprogramme rufen die von DOS bereitgestellten Funktionen mit Hilfe von Interrupts auf. OS/2 stellt DOS-Anwendungen diese DOS-Interrupts zur Verfügung. Der Maschinencode einer DOS-Anwendung läuft also direkt ab, während alle Aufrufe von DOS-Interrupts letztendlich in OS/2-Funkionen münden.

OS/2 und Windows definieren jeweils Schnittstellen für Anwendungsprogramme (Application Programming Interface, API), bestehend aus einem Satz von Funktionen, Datenstrukturen, konstanten Werten und Nachrichten für Fenster. Windows (bis zur Version 3.11) ist im strengen Sinne kein Betriebssystem, sondern eher als ein grafischer Aufsatz auf DOS anzusehen. Aus Sicht des Programmierers ist es jedoch unerheblich, ob das benutzte API Teil des Betriebssystems ist oder eine separate Komponente darstellt, solange beide in der Laufzeitumgebung vorhanden sind. Windows 95 ist eigentlich nichts anderes als eine engere Integration und Weiterentwicklung des Zweiergespannes DOS und Windows. Dagegen ist Windows NT ein komplett neu entwickeltes Betriebssystem, das Windows-Anwendungen den (nahezu) gleichen Satz von Windows-APIs und DOS-Anwendungen die benötigten DOS-Interrupts bereitstellt.

Mit den verschiedenen Windows-Versionen hat sich auch das API ständig weiterentwickelt. Während Windows bis zur Version 3.11 ein 16-Bit-API bereitstellte, bieten Windows NT und Windows 95 32-Bit-APIs an, wofür der Oberbegriff »Win32« verwendet wird, auch wenn die APIs nicht vollständig identisch sind. Damit neue 32-Bit-Software auch unter Windows 3.11 läuft, bietet Microsoft ein eingeschränktes 32-Bit-Subsystem für 3.11 an und definiert eine entsprechende API-Untermenge: »Win32s«. Wesentliche Merkmale von 32-Bit-Betriebssystemen, wie zum Beispiel Multithreading, kann aber eine Software, die sich auf Win32s beschränkt, nicht ausnutzen.

WIN-OS/2 bietet die binäre Portabilität von 16-Bit-Windows-Anwendungen unter OS/2. Zusätzlich unterstützt es das Win32s-Subsystem. Von diesem gibt es allerdings verschiedene Versionen. So bietet Win32s ab der Version 1.25a sogar OLE-Unterstützung. Demgegenüber bietet die aktuelle Version 1.30 nur noch geringe Zusatzmerkmale. So erwähnt die aktuelle SDK-Dokumentation (MSDN Library July/1996) ausschließlich die Version 1.25. Jedoch greift das aktuelle Subsystem nun auf Speicherbereiche zu, die OS/2 nicht zur Verfügung stellen kann. OS/2 Warp V3 wurde mit der Unterstützung von Win32s 1.15 ausgeliefert, kann aber für die Version 1.25a mit OLE nachgerüstet werden. OS/2 Warp 4.0 bietet dies direkt an.

7.5.3 Portable Entwicklung für OS/2 und Windows – Viele goldene Wege

Die meisten Softwarefirmen heutzutage bieten Ihre PC-Anwendungen auf dem großen Markt der Windows-Plattformen an. Die Entscheidung dieser Firmen, ihre Software nicht auch

zusätzlich auf OS/2 anzubieten, beruht auf der Annahme, daß die Kosten, um beide Plattformen mit zwei Codebasen zu unterstützen, den zusätzlichen Gewinn im OS/2-Markt übersteigen.

Dabei ist es durchaus möglich, beide Plattformen mit einer Codebasis abzudecken und somit die Kosten dramatisch zu verringern. Hierfür gibt es eine Reihe von Schnittstellen, Produkten und Technologien, die dies gerade für OS/2 und Windows ermöglichen. Hierzu zählen nicht nur die erwähnten IBM-Lösungen, sondern auch Produkte anderer Hersteller.

So stehen Compiler und Interpreter für nahezu jede Programmiersprache unter OS/2 zur Verfügung. Die im folgenden aufgeführten Werkzeuge sind entweder auf beiden Plattformen verfügbar oder bieten eine hohe Kompatibilität zu einem entsprechenden Windows-Werkzeug.

Programmiersprache	**IBM VisualAge**	**Anderer Hersteller**
Java	... for Java	Sprache ist binär portabel
C/C++	... for C++	Borland C++ (Borland)
Smalltalk	... for Smalltalk	Parcplace Smalltalk (Parcplace)
Cobol	... for Cobol	MicroFocus Cobol (MicroFocus)
VisualBasic	... for Basic	–
Pascal	–	SpeedPascal (Speedsoft), Virtual Pascal (fPrint) (beide kompatibel zu Turbo Pascal)
Xbase	–	Xbase/2 (Alaska)

Bibliotheken für die portable Entwicklung der Benutzerschnittstelle liegen den C++-Compilern von IBM (IBM Open Class) und Borland (OWL) bei. Zum anderen unterstützen die meisten separat erhältlichen Bibliotheken OS/2, wie zum Beispiel »StarView«, »XVT«, »zApp« und »Zinc«.

Die Entwicklung portabler Software gesellt sich zusätzlich zu den derzeitigen Herausforderungen an Windows-Entwickler:

- Die Vorteile der Objektorientierung rechtfertigen mehr und mehr den Aufwand des Umstiegs von der prozeduralen Programmierung.
- Der Boom von Windows 95 und Windows NT macht die Portierung von 16-Bit-Windows-Anwendungen auf 32 Bit notwendig.
- Die Einbindung neuer Technologien wie zum Beispiel Internet und Spracherkennung wird vom Markt erwartet.

Hierbei hat die Entscheidung, ob konventionell oder objekt-orientiert weiterentwickelt wird, die größte Tragweite. Daher gibt es unterschiedliche Unterstützungspfade für die portable Entwicklung für Windows und OS/2 (siehe Abbildung).

Entscheiden Sie sich für die objektorientierte Entwicklung, bietet IBM Ihnen die VisualAge-Familie, IBM OpenClass und SOM für die portable Entwicklung an. Bleiben Sie weiterhin bei der konventionellen Entwicklung, kommen Open32 und SMART in Frage. Hierbei bietet sich der einmalige Aufwand an, beim Umstieg von 16-Bit- auf 32-Bit-Windows auch gleichzeitig die OS/2-Portabilität mit Open32 in die Änderungen mit einzubeziehen.

7.5.4 OS/2 und Windows – Verwandte ersten Grades

Die OS/2-Entwicklung war ursprünglich eine Zusammenarbeit von IBM und Microsoft. Dahinter stand die Absicht, das Nachfolgebetriebssystem für DOS herauszubringen. In dieser Zeit entwickelte Microsoft gleichzeitig auch eine grafische Benutzeroberfläche für DOS. Jedoch war dem 1985 erscheinenden Windows 1.0 wenig Erfolg beschieden. Dieser stellte sich erst mit der dritten Version ein.

Viele der Ideen und Entwicklungen fanden sowohl in Windows als auch OS/2 Eingang, und beide haben viele ähnliche Architekturen. So bieten beide zum Beispiel

– ein hierarchisches Fenstersystem,

– eine ereignisgesteuerte Nachrichtenschleife,

– separate Subsysteme für Grafik und Drucken,

– nahezu identische Ressourcen-Definitionen für Fenster, Dialoge, Icons etc.,

– sehr viele ähnliche API-Funktionen,

– dynamisches Binden von Bibliotheken (DLLs).

In Anbetracht dieser Verwandtschaft und der Ähnlichkeit beider Systeme wird es verständlich, daß Open32 in der langen Geschichte von OS/2 mittlerweile die vierte (!) Portierungsunterstützung für Entwickler darstellt.

Das erste Werkzeug hieß »Windows Libraries for OS/2« (WLO) und kam von Microsoft, das zu jenem Zeitpunkt Entwicklern helfen wollte, ihre Produkte auch unter OS/2 anzubieten. Microsoft konvertierte den »Windows Help Manager« mit WLO, so daß auch das Hilfesystem auf beiden Plattformen verfügbar war. WLO fand insgesamt keinen großen Verbreitungsgrad. Wenn man aber im Internet bei Microsoft nach WLO sucht, findet man immerhin einige Fehlerbeschreibungen, die aufzeigen, daß das Produkt »Microsoft Mail for PC Networks, OS/2 Workstation« mit Hilfe von WLO auf OS/2 portiert wurde.

WLO war nichts anderes als eine Bibliothek, mit der die Windows-Anwendung gebunden wurde. Dabei wurde alles, auch die Ressourcen, während der Laufzeit konvertiert. Es handelte sich dabei also um eine Art Windows-Emulator unter OS/2.

Die Version 2.0 von OS/2 brachte signifikante Designänderungen:

– Die SOM-basierende Arbeitsoberfläche als neue intuitive Benutzerschnittstelle

– Die Speicherverwaltung benützte nun den linearen Adreßraum des 386, statt wie bisher die Segmentierung des 286.

– Die DOS-Unterstützung wurde mit einem besseren Konzept implementiert.

Für OS/2 2.0 wurde 1992 ein Portierungswerkzeug namens »Mirrors« von der Firma Micrografx entwickelt, das die Portierung von Windows-3.0-Anwendungen unterstützte. Ähnlich wie WLO bestand auch »Mirrors« aus einer DLL, mit der die Windows-Anwendung gebunden wurde. Jedoch arbeitete die »mirrors.dll« eine Reihe der Windows-Funktionen intern ab und leitete nur einen Teil an OS/2 weiter. Das Mirrors-Toolkit enthielt zusätzlich einen Ressourcen-Compiler, der die Windows-Ressourcedateien direkt in OS/2-Ressourcen umsetzte und ein Konvertierungswerkzeug namens »MirHelp« für die Hilfedateien. So mußten Ressourcen und Hilfe nicht während der Laufzeit umgesetzt werden. Mirrors wurde durchaus für die Portierung größerer Projekte eingesetzt, zum Beispiel bei der OS/2-Version von WordPerfect. Falls Sie interessiert, was aus Mirrors geworden ist, suchen Sie mal im Verzeichnis »\os2\dll« Ihres OS/2-Laufwerks nach der Datei »mirrors.dll«. Die Laufzeitunterstützung ist also weiterhin gegeben, auch wenn das Toolkit nicht mehr erhältlich ist.

Einen völlig anderen Weg beschreitet »SMART« von der Firma One-Up. Es dient der halbautomatischen Umsetzung der gesamten Codebasis, also Quelltexte, Ressourcendateien, Hilfetexte und Ressourcen von Windows zu OS/2 zu konvertieren. Zusätzlich analysiert es die Codebasis und ermöglicht dadurch vorab eine Abschätzung des Portieraufwands. In einem zweiten Schritt wird dann die komplette Codebasis zum großen Teil automatisch übersetzt. Ein Programmiereditor mit speziellem Hyperlinking zwischen den zu bearbeitenden Dateien erleichtert die restliche Handarbeit. Die Onlinehilfe erklärt für jedes API-Element, welches OS/2-Äquivalent es gibt und welche Besonderheiten unter OS/2 zu beachten sind. Die folgende Abbildung zeigt die unterschiedlichen Komponenten von SMART auf.

Die Umsetzung des Quelltexts, C- oder C++-Code, wird durch eine primäre Umsetzungstabelle gesteuert. Fünf Tabellen stehen zur Verfügung:

Windows 16-Bit	–>	OS/2 32-Bit
Windows 32-Bit	–>	OS/2 32-Bit
OS/2 16-Bit	–>	OS/2 32-Bit
Windows 16-Bit	–>	OS/2 32-Bit mit Open32
Windows 32-Bit	–>	Open32

Die Tabellen führen zu jedem API-Element der jeweiligen Quellplattform auf, wie portabel dieses Element ist und gegebenenfalls das entsprechende OS/2-Äquivalent. Der automatisierte Prozeß kann noch erheblich verbessert werden, indem man eine zweite Tabelle (User Defined Migration Database, UDMD) erstellt, die weitere APIs erkennt und komplexere Umsetzungen in der »Migration Command Language«, eine einfache, Rexx-ähnliche Sprache, implementiert.

Während der Ansatz von SMART sehr mächtig ist, hat er auch einen gravierenden Nachteil. Sie haben nun nicht mehr eine Codebasis, sondern eine zweite für OS/2. Bei der Weiterentwicklung Ihrer Software müssen Sie also entweder zwei Codebasen pflegen, oder den Umsetzungsaufwand bei jeder neuen Version wieder auf sich nehmen und Ihre Software entsprechend zeitversetzt auf OS/2 anbieten. Genau hier setzt Open32 an.

7.5.5 Open32 – Ein »offenes« Win32

Im Mai 1995 kündigte IBM die Developer API Extensions (DAPIE) [4] an, die nun als Open32-API in OS/2 Warp 4.0 eingegangen sind. Die rund 750 Funktionen und 300 Nachrichten stellen eine Auswahl des Win32-APIs dar, die anhand der Analyse von rund neun Millionen Zeilen Code vorgenommen wurde, unter anderem aus der Lotus SmartSuite. Dadurch erklärt sich auch die Größe der gemeinsamen Codebasis, wie sie von IBM in diesem Papier eingeschätzt wird.

Der Ansatz von Open32 unterscheidet sich wiederum von den drei bisherigen – WLO war ein Emulator, der die Umsetzung während der Laufzeit vornahm. Mirrors war zum Teil ein Emulator und zum Teil wurde die Umsetzung zur Zeit des Bindens durchgeführt. SMART bietet keinerlei Emulationsschicht, sondern unterstützt die Umsetzung der Quelltexte zur Entwicklungszeit.

Auf den ersten Blick gleicht Open32 dem Konzept von Mirrors. Die Laufzeitumgebung von Open32 stellt dabei die »pmwinx.dll«, »pmddeml.dll« und einige Reg-Dateien für die Registry bereit. Allerdings sind diese nicht etwa ein Aufsatz auf dem bisherigen Presentation Manager, sondern sind in diesen integriert.

Dadurch können Open32-Anwendungen weitere OS/2-Schnittstellen benützen und somit zusätzliche Funktionalität anbieten. IBM hat für diese Integration auch einige Änderungen in OS/2 Warp vorgenommen:

– Die Open32 Registry wird auch allen sonstigen OS/2 Anwendungen zur Verfügung stehen

– Der Ressource-Compiler des OS/2 Warp Toolkit

– WinSocks

Seit OS/2 2.0, Windows 3.0 und Mirrors sind Anzahl und Umfang der APIs in beiden Betriebssystemen erheblich erweitert worden. Daher ist es verständlich, daß Open32 bewußt die gebräuchlichsten Funktionen und Nachrichten aus dem eigentlichen Kern des Win32-APIs bereitstellt. Wenn Sie ihre Anwendung auf OS/2 portieren, müssen Sie deshalb nicht auf die neuen Funktionen verzichten, in den meisten Fällen gibt es die entsprechende Funktionalität unter OS/2.

Allerdings sind diese Schnittstellen inkompatibel und diese Teile Ihrer Anwendung müssen Sie doppelt entwickeln und pflegen. Die folgende Tabelle gibt Ihnen einen Überblick über die Windows-Schnittstellen, die über Win32 hinausgehen, und ihre Entsprechungen unter OS/2 Warp beziehungsweise Zusatzprodukte, die dies ermöglichen. Sie enthält die in der MSDN-Library Juli/1996 als SDK aufgeführten Schnittstellen. Spezielle OS/2-Warp-4.0-Schnittstel-

len, die nicht unter Windows verfügbar sind, sind für die Portierung von Windows-Anwendungen irrelevant und werden nicht erwähnt.

Anwendungsbereich	**Windows-API**	**OS/2 API oder Komponente**
Cartoons	WinToon	–
Datenbankzugriff	ODBC	INTERSOLV ODBC **
	DAO	–
Spiele	WinG	DIVE, EnDIVE
Sprache	Speech Recognition API	VoiceType *

Die folgende Tabelle zeigt die zahlreichen Erweiterungen zum Win32-SDK und wiederum die Entsprechungen unter OS/2 Warp.

Funktion	**Windows**	**Warp**
Unbestätige Nachrichten	Mailslots (3 Funktionen)	NetBIOS (1 Funktion)
Netzwerkressourcen	WNet	vorhanden
Netzdatenaustausch	NetDDE	NetDDE
Performance Monitor	Performance Monitoring	vorhanden
Services	Service Control Manager	–
Sicherheit	Security	SES *
LAN Server	LAN Manager API	LAN Server API
Logmechanismen	Event Logging	FFST
Fernzugriff	RAS	LAN Distance
Komponentenarchitektur	OLE	OpenDoc
Audiozugriff		
RPC	MS RPC	TCP/IP RPC
Multimedia	Multimedia	MMPM/2
3D-Grafik	OpenGL	OpenGL *
Sockets	WinSocks	TCP/IP Sockets
	–	WinSocks *
Lizenzierung	LSAPI	–
E-Mail	MAPI	Lotus Notes Mail **
Telephonie	TAPI	–
Stiftunterstützung	Pen API	Pen SDK
Direkter Videozugriff	DirectX	EnDIVE
Ausnahmebehandlung	Structured Exception Handling	C- bzw. C++-Standard
Datenaustausch	DDE	DDE
	DDEML	DDEML *

Dateien im Speicher	Memory mapped files	Shared memory
Pipes	Pipes	Pipes
Registrierungsinfo	Registry	Profile
	–	Registry *
Erweiterte Controls	Common Controls Library	Advanced Controls

* *in OS/2 Warp 4.0 neu integriert, in den meisten Fällen aber auch für OS/2 Warp separat erhältlich.*

** *zusätzliches Produkt, Notes Mail ist OS/2 Warp 4.0 beigelegt.*

Interessant ist hierbei auch, daß sich gerade im Anwendungsbereich Intranet die OS/2-Schnittstellen stärker an Standards (DCE, TCP/IP) oder weitverbreiteten Implementierungen unter Unix (Sun RPC) halten. Während die Windows-APIs davon abweichen und mit Windows-spezifischen, sprich inkompatiblen, Elementen erweitert sind (WinSocks, MS RPC).

Anhand dieser Gegenüberstellung wird deutlich, daß das Open32-API allein in den wenigsten Fällen ausreicht, jedoch in der Regel Entsprechungen existieren. Wichtig ist jedoch, daß mit Open32 die meist genutzten Kernfunktionen des APIs unterstützt werden. Dies umfaßt die folgenden Bereiche:

Basisfunktionen des Betriebssystems:

– Atome

– Datum und Zeit

– Umgebungsvariablen

– Speicherverwaltung

– Modulverwaltung

– Drucken

– Prozesse und Threads

– Registry

– Dateiverwaltung

Bereitgestellte Dialogboxen

– Auswahl von Farbe und Schriftsatz

– Öffnen und Speichern von Dateien

– Drucken

– Finden und Ersetzen von Text

Schnittstelle zu Grafikgeräten

– Bitmaps, Pinsel, Stifte
– Farben und Farbpalette
– Schriftsätze und Text
– Linien, Kurven, Rechtecke, Polygone, Ellipsen, Chords
– Pfade und Regionen
– Umwandlungsmodi, Transformationen, Koordinaten

Fensterverwaltung

– Accelerators, Carets, Zeiger, Icons, Menüs, Stringtabellen
– Fenster und Dialogboxen
– Knöpfe, Kombiboxen, Editfelder, Listboxen, Schiebebalken
– Nachrichten und Nachrichtenschlangen
– Rechtecke
– Zeitgeber
– Dynamischer Datenaustausch (DDE und DDEML
– Mehrdokumentenschnittstelle (MDI)

7.5.6 Aller Anfang ist Installation

Für die Anwendungsentwicklung unter OS/2 brauchen Sie mindestens einen C-Compiler und das Developer's Toolkit for OS/2. Das Open32-API ist in der neuesten Version für OS/2 Warp 4.0 enthalten. Sie benötigen außerdem SMART zur Codeanalyse und zur automatischen Portierung der Ressourcen, Ressource- und Hilfedateien. Für die plattformunabhängige Entwicklung Ihrer Hilfe- und Online-Dokumente können Sie zusätzlich den WYSIWIG-Editor Hyperwise einsetzen. Falls Sie unter OS/2 Warp 3 entwickeln, müssen Sie FixPak 17 (oder besser höher) und die Laufzeitumgebung für Open32 einspielen.

All diese Komponenten befinden sich auf der neuesten Version (#11) der IBM Developer Connection (Die IBM Developer Connection ist ein CD-Abonnenment für OS/2 und AIX , das etwa mit dem MS Developer Network-Professional vergleichbar ist. Sie können diese über 0130/812177 bestellen.) inklusive zeitlimitierter Versionen von VisualAge for C++ for OS/2, OS/2 Warp 3.0 und OS/2 Warp 4.0 (ohne WIN-OS/2). Nachdem Sie die Developer-Connection-CD für OS/2 installiert haben, lassen sich alle weiteren Komponenten mit Hilfe des »DevCon Catalog« leicht installieren. Für die konventionelle Entwicklung mit C und Open32 können Sie zahlreiche Komponenten des Developer's Toolkits und VisualAge for C++ weglassen. Zum Beispiel enthält der Compiler das komplette Toolkit in der älteren Version 3.0.

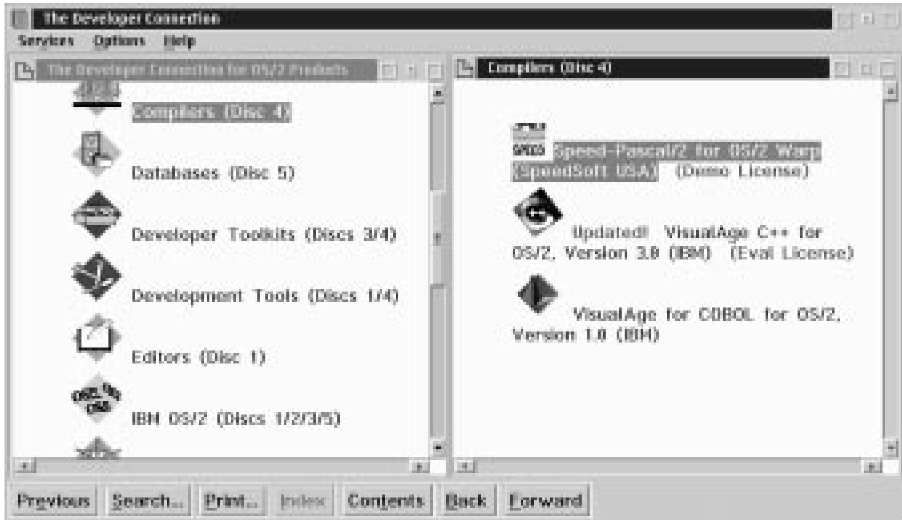


Abb. 7.29: Katalog der IBM Developer Connection for OS/2

Der Speicherbedarf des Toolkits reduziert sich von 117,4 Mbyte auf 16 Mbyte, wenn Sie die benötigten und hier empfohlenen Komponenten installieren:

– IBM Developer's Toolkit for OS/2 Warp 4.0

– Development Tools

– Base Tools

– Toolkit Information

– Control Program Guide and Reference (empfohlen)

– Graphics Program Guide and Reference (empfohlen)

– Presentation Manager Guide and Reference (empfohlen)

– Open32 Programming Guide and Reference (empfohlen)

– Headers and Libraries

– C Runtime Headers

– OS/2, PM & MM C/C++ Headers

– Libraries

– Sample Programs

– Open32 Samples (empfohlen)

– Visual Age for C++

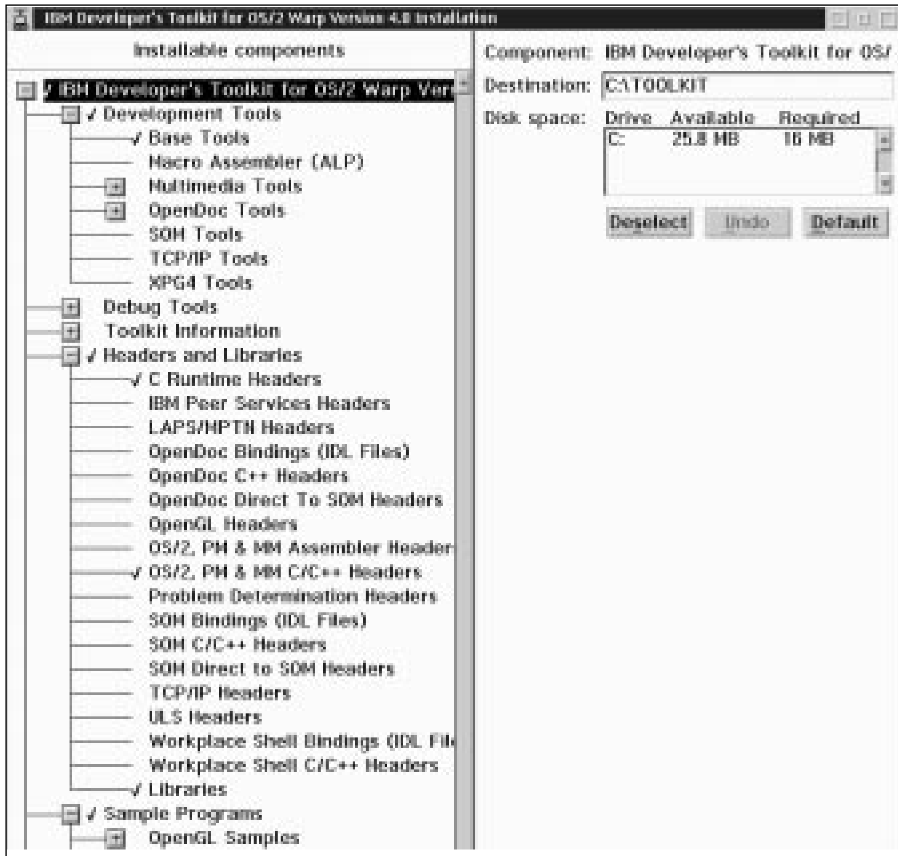


Abb. 7.30: Benötigte Komponenten des Developer's Toolkit for OS/2 Warp, Version 4.0

Auch der Platzbedarf von VisualAge for C++ verringert sich von 181 Mbyte auf 60 Mbyte, wenn Sie nur die folgenden Komponenten installieren:

– Compiler and Runtime Libraries

– Workframe (empfohlen)

– PM Debugger (empfohlen)

– Documentation (empfohlen)



Abb. 7.31: Benötigte Komponenten von VisualAge for C++ for OS/2

Achten Sie nach der Installation darauf, daß der neue Ressourcen-Compiler aus dem Toolkit im Suchpfad als erstes gefunden wird.

SMART liefert eine Zusatztabelle für die Analyse von Windows 32-Bit-Code auf Open32-API. Diese liegt im Quelltext vor. Vor der Verwendung müssen Sie diese erst umwandeln:

1. Starten Sie »SMART2 Source Migration« aus dem »SMART2 Toolset«-Ordner.
2. Wählen Sie den Menüpunkt »Create User UDMD...« aus dem Menü »Tables«.
3. Selektieren Sie als »Directory for User UDMD Migration Data Base« das Unterverzeichnis »SmartAux\Dapie32« Ihres SMART-Installationsverzeichnisses.
4. Selektieren Sie als »Input Dictionary Filename« die Datei »dapie32.dct« aus dem gegebenen Verzeichnis.
5. Starten Sie die Umwandlung mit dem Druckknopf »Process«

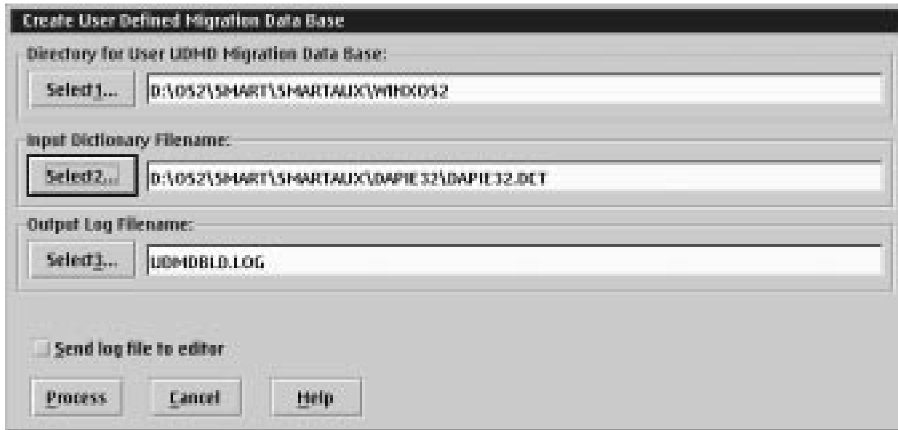


Abb. 7.32: Umwandlung der »Win32 to DAX«-Zusatztabelle

7.5.7 Designbetrachtungen

Die üblichen Designbetrachtungen für traditionelle Anwendungsprogramme gelten natürlich auch für portable Anwendungen. Jedoch sollten weitere Empfehlungen beachtet werden, um den Grad der Portabilität möglichst zu maximieren. Sie finden solche Empfehlungen in der Online-Dokumentation des Developer's Toolkit [5] und von SMART [6]. Eine Reihe von Büchern sind über die Entwicklung für mehrere Plattformen [7] und über C- und C++-Portabilität [8] geschrieben worden. Informationen zur Portabilität zwischen OS/2 und Windows bieten IBM [1] und Microsoft [9] an. Speziell auf Open32 geht ein neues IBM Redbook [10] ein.

Die folgende Liste gibt einen kurzen Überblick über die wesentlichem Empfehlungen und Tips. Sie ersetzt nicht die genannten Referenzen während der Entwicklung portabler Anwendungen.

Allgemein

- Isolieren Sie nicht portablen Code in separaten Quelltexten.
- Verwenden Sie bedingte Compilierung. Bei VisualAge for C++ können Sie die vordefinierten Makros für die jeweilige Plattform verwenden: »__OS2__« für OS/2 und »__WINDOWS__« für Windows.
- Kommentieren Sie die Gründe für nicht-portable Elemente im Quelltext.
- Größe, Alphabet und Beachtung der Groß/Kleinschreibung für Bezeichner (z.B. Dateinamen) können auf verschiedenen Plattformen variieren.
- Nicht alle Plattformen unterstützen Multitasking und Multithreading.
- Beachten Sie die Folgen, wenn Sie Bibliotheken von Drittherstellern verwenden.
- Verwenden Sie möglichst Compiler und Werkzeuge, die auf allen gewünschten Zielplattformen vorhanden sind.

C und C++

- Beschränken Sie sich auf die beschriebenen ANSI-/ISO-Standards [11, 12].
- Größe von C-Datentypen und Ausrichtung von Strukturen und Klassen können unterschiedlich sein.
- Unterschiedliche Gültigkeitsbereiche bei numerischen Typen sind möglich.
- Verwenden Sie Prototypen für alle Funktionen.
- Beschränken Sie sich möglichst auf die C/C++-Standardbibliotheken für Speicherverwaltung, Platten- und Dateizugriff und Prozeßverwaltung. Sofern diese Bibliotheken nicht ausreichen, isolieren Sie diese Funktionalität in separaten Quellcode als Makros oder Inline-Funktionen.
- Verlassen Sie sich nicht darauf, daß »int« eine bestimmte Größe oder die selbe wie »long int« oder »short int« hat.

OS/2 und Windows

- Die Tabelle spezieller Tastenkombinationen (Accelerators) wird unterschiedlich verarbeitet. Außerdem wirken sich Groß-/Kleinschreibung und die Hochstelltaste in dieser Tabelle unterschiedlich aus.
- Der Koordinatenbereich differiert zwischen OS/2, Windows 95 und NT.
- Das Exportieren und Importieren von Funktionen aus DLLs ist unter Windows und OS/2 etwas unterschiedlich gelöst.
- Bis zur Version 3.51 von Windows NT war es möglich, auf dem Entwicklungsrechner sowohl OS/2 als auch Windows mit dem HPFS-Dateisystem zu verwenden, so daß lange Dateinamen aus beiden Plattformen benützt werden konnte. Microsoft unterstützt mit dem neuen Windows NT 4.0 nicht mehr das HPFS-Dateisystem. Als einziger gemeinsamer Nenner der unter OS/2 und Windows existierenden Dateisystemen (FAT, VFAT, HPFS, NTFS) verbleibt nur noch das FAT-Dateisystem, das jedoch keine langen Dateinamen unterstützt. Falls Sie ein verläßliches Netzwerk etabliert haben, können Sie Ihre Projektdateien im Netzwerk speichern. Die LAN-Dateisysteme von OS/2, Windows 95 und NT unterstützen lange Dateinamen transparent.

Presentation Manager API

- OS/2 und Windows Koordinatensystem unterscheiden sich in der Verwendung der Y-Achse. Setzen Sie Makros zur Berechnung der y-Werte ein.
- Verwenden Sie Makros für die Interpretation auf die Nachrichtenparameter. OS/2 und Windows definieren die Parameter unterschiedlich.
- Benützen Sie Makros für das Verschicken und die Formatierung von Nachrichten.
- Die Funktionen für Grafikprimitiven und -attribute unterscheiden sich geringfügig. Isolieren Sie diese in separaten Modulen.

Open32-API

– Die Registry benützt andere Wurzelverzeichnisse.

– Open32 greift direkt auf echte OS/2-Ressourcen zu. Die Open32-Ressource-Funktionen benützen entsprechend die Datenstrukturen für OS/2 und nicht Win32-Ressourcen.

– Open32- und OS/2-Handles sind nicht austauschbar, d.h., ein Handle, das durch eine Open32-Funktion erzeugt wurde, sollte nicht an eine OS/2-Funktion weitergegeben werden. Als wichtige Ausnahme können Sie OS/2-Fenster innerhalb von Open32-Fenstern und umgekehrt erzeugen.

– Mischen Sie nicht OS/2- und Open32-Funktionen des selben Anwendungsbereiches wie zum Beispiel des Grafiksubsystems (GDI).

7.5.8 Portierung einer Windows-Anwendung

Im folgenden werden Sie eines der mit dem Developer's Toolkit mitgelieferten Open32-Beispiele Schritt für Schritt auf Open32 portieren. Alle mitgelieferten Beispiele sind einfache Win32-Demoanwendungen, die komplett mit Open32 portiert werden können. Wenn Sie Ihre Anwendungen mit SMART und Open32 portieren, werden Sie die hier aufgeführte Vorgehensweise wählen. Die kursiv gekennzeichneten Schritte sind nicht auf das Beispiel anwendbar und deshalb nicht näher beschrieben.

1. Anpassen der Projektstruktur

2. Analyse der Codebasis

3. Modifikation der Codebasis durch Ersetzen nicht von Open32 unterstützter APIs, Separieren der plattformspezifischen Codeteile und wiederholter Codeanalyse

4. Analyse der Ressourcen-Dateien

5. Anpassung der gemeinsamen Codebasis

6. SMART-Portierung oder Neuentwicklung der plattformspezifischen Codeteile für OS/2

7. Portierung der Hilfedateien

8. Bauen der OS/2-Anwendung

Anpassen der Projektstruktur

Kopieren Sie das Open32-Beispiel »Browser« vom Unterverzeichnis Samples\Open32\Browser Ihres Developer's-Toolkit-Installationsverzeichnisses in ein neues Projektverzeichnis \Prj\Browser. Richten Sie in diesem Projektverzeichnis ein Unterverzeichnis »OS2« für die OS/2 spezifischen Dateien ein, und ein Unterverzeichnis »Analyse« für die Analyseinformationen.



Abb. 7.33: Das Projektverzeichnis für Browser

Im Fall Ihrer realen Anwendung erzeugen Sie ein weiteres Unterverzeichnis »Win« für die Windows-spezifischen Dateien. Die Windows-spezifischen Dateien »browser.rc« und »icon1.ico« werden während des Baus der OS/2-Anwendung automatisch umgesetzt und verbleiben deshalb im gemeinsamen Hauptverzeichnis. Die Projektstruktur komplexer Anwendungen besteht in der Regel aus mehreren Unterverzeichnissen, die hier vorgegebene Strukturierung kann entsprechend auf weitere Unterverzeichnisse angewendet werden.

Analyse der Codebasis

Bevor Sie mit SMART eine Analyse durchführen oder Code umwandeln, erstellen Sie eine Datei, die die Liste aller relevanten Dateien enthält. Anhand dieser Listendatei können Sie dann die weiteren Schritte durchführen. Erstellen Sie diese Datei im »Analyse«-Unterverzeichnis, so daß SMART die Ergebnisdateien entsprechend auch dort ablegt.

1. Starten Sie »SMART2 Source Migration« aus dem »SMART2 Toolset«-Ordner

2. Wählen Sie *Maintain a List-of-Files...* aus dem *File*-Menü

3. Geben Sie im Dialog *Select File for List-of-Files* das Verzeichnis »\prj\browser\analyse« und den Dateinamen »code.lst« an.

4. Bestätigen Sie die Abfrage, ob diese nicht existierende Datei wirklich erzeugt werden soll.

5. Im Dialog *Files List* drücken Sie den Druckknopf *Add...*, um die Quelltexte hinzuzufügen.

6. Im Dialog *Add Files To List* wählen Sie das Projektverzeichnis »\prj\browser«, selektieren die Dateien »browser.c« und »browser.h«, drücken auf *Add*, um diese Dateien in die Listendatei hinzuzufügen, und *Close*, um diesen Dialog zu beenden.

7. Im Dialog *Files List* drücken Sie auf *Save*, um die Listendatei abzuspeichern.

Während diese Vorgehensweise mit einer Listendatei bei einem Beispiel mit wenigen Quelltexten aufwendig erscheint, helfen die verwendeten Dialoge im Realfall ähnlich schnell, eine Listendatei mit exakt den interessierenden Dateien zu erstellen. Die weiteren Schritte sind dann unabhängig vom Umfang der Codebasis, außer das Analyse und Migration einer großen Codebasis recht zeitintensiv ist.

Damit Sie einen Eindruck erhalten, welche Einsparung Ihnen Open32 bietet, führen Sie drei Analysen durch. Mit der ersten Analyse schätzen Sie den Aufwand für eine direkte Portierung auf OS/2 Presentation Manager ein. Bei der zweiten verwenden Sie eine Zusatztabelle für die mit SMART mitgelieferte MDI-Unterstützung unter OS/2. Die dritte Analyse betrachtet schließlich die Verwendung des Open32-APIs. Beachten Sie, daß die angegebenen Tabellen für Win32-Code gedacht sind, für 16-Bit-Windows-Code bietet SMART andere Tabellen an.

1. Wählen Sie *Select Tables...* aus dem Menü *Tables*.

2. Im Dialog *Migration Tables* führen Sie die folgenden Schritte durch:

– Selektieren Sie als *SMART Primary Table Filename* die Tabelle »tables\win32os2\win32os2.tbl«.

– Drücken Sie auf den Auswahlknopf *Do Not Use UDMD Table*.

– Drücken Sie auf *Set*, um diese Einstellungen abzuspeichern.

3. Wählen Sie *Analyze Source Code..* aus dem Menü *Analysis*.

4. Für die reine Analyse reichen die vordefinierten Auswahlpunkte. Sie können also einfach den Button *Process* drücken, um die Analyse durchzuführen.

SMART zeigt den Analysebericht an. Der Bericht gibt Aufschluß über Codeumfang in Byte, Zeilen und Codezeilen. Übrigens ist das Delta zwischen Codezeilen und Gesamtzeilen ein guter Hinweis darauf, wie gut die Codebasis kommentiert ist! In der Spalte *Hits* ist die Anzahl aufgeführt, wie oft Elemente des Win32-APIs im Code auftauchen. Die nächste Spalte setzt dies ins Verhältnis zur Gesamtanzahl an Codezeilen. Die Spalte *Instances* gibt die Anzahl der entdeckten Win32-API-Elemente an. Die Spalte *Effort* gibt den geschätzten Portierungsaufwand in Stunden an. Sie beruht auf Erfahrungswerten von »One Up« mit der Voraussetzung, daß die Portierung von einem durchschnittlichen Windows-Entwickler ohne OS/2-Kenntnisse durchgeführt wird. Mit dem Eingabefeld *Effort Modifier* im Dialog *Source Code Analysis* läßt sich dies an die jeweiligen Gegebenheiten anpassen.

SMART2 Source Migration - (E:\PRJ\BROWSER\ANALYSE\CODE.LST)

File Analysis Migration Resources Tables Options Help

Path/Filename	Size	Lines	Code	Hits	% Hits	Instances	% Inst.	Effort
E:\PRJ\BROWSER\								
BROWSER.C	36,350	759	562	341	60.7	146	26.0	105.9
BROWSER.H	985	26	16	0	.0	0	.0	.0
SubTotal (2)	37,335	785	578	341	59.0			105.9
Total (2)	37,335	785	578	341	59.0	146	25.3	105.9

Abb 7.34: SMART-Analysebericht für reine OS/2-Portierung

Zusätzlich speichert SMART eine Zusammenfassung der Aufwandsersparnis in der Datei »code.sum« und eine ausführliche Aufstellung der gefundenen Elemente in »code.rpt«. In der Aufwandsabschätzung wird deutlich, daß SMART bei diesem Beispiel nur 23,2 Prozent Ersparnis einbringt.

```
Migration Analysis Report Summary

Total File Count                                             2
Total File Size (Bytes)                                 37,335

Total Source Lines                                         785
Total Code Lines                                           578

Total Category 000 Hits                                     34
Total Category 010 Hits                                     66
Total Category 020 Hits                                    108
Total Category 030 Hits                                     60
Total Category 040 Hits                                     34
Total Category 050 Hits                                     22
Total Category 999 Hits                                     17
Total Migration Hits                                       341

Total Cat 000 & 010 Hits Auto Migrated                     100
Total Hits MiCL Processed                                    0
Total Hits Processed with UDMD                               0
Total Hits Excluded                                          0

Total Migration Keyword Instances                          146

Estimated Migration Effort Without SMART                 131.0
Savings from SMART Base Migration                         14.2

Savings from Cat 010 Auto Migrate                         16.2
Savings from Keyword Exclusions                             .0

** Net Migration Effort **                               100.6 Percent Effort
Reduction using SMART                                 23.2 %
```

Abb. 7.35: SMART-Aufwandsersparnis für reine OS/2-Portierung

Im Menü *Analysis* können Sie mit den Menüpunkten *Display Recap...* und *Display Detail..* zwischen der Ansicht von ausführlicher Aufstellung und Bericht wechseln. Das nächste Listing ist ein Auszug aus der Datei »code.rpt«. Die Unterteilung in Funktionsbereiche in der ausführlichen Aufstellung gibt Hinweise auf Teilbereiche des APIs, in denen besonderer Aufwand nötig ist. Übrigens sollten Sie sich nicht durch die Kategorie »Functionality does not exist« verunsichern lassen. Zum Beispiel gibt es die Struktur PAINTSTRUCT nicht unter OS/2, sie wird allerdings hier auch nicht benötigt. Solche vergleichende Informationen finden Sie übrigens im »SMART2 Viewer«.

	ITEMS	FILES INSTS	PCT INSTS	EFFORT
CATEGORIES				
000 Informational only	1	10	6.8	.0
010 Literal replacement	1	33	22.6	0
020 Replacement with parameter changes	1	52	35.6	20.2
030 Change with more / less API calls	1	22	15.1	17.0
040 Logic changes required	1	10	6.8	14.5
050 Functionality does not exist	1	9	6.2	21.6
999 Dictionary entry not defined	1	10	6.8	7.1
*** Category Totals	2	146	100.0	80.4
TYPES				
010 Function	1	62	42.5	45.7
020 Message	1	13	8.9	4.5
030 Symbol	1	49	33.6	15.5
040 typedef	1	22	15.1	14.7
AREAS				
030 Memory-Management	1	10	6.8	2.3
060 Resource-Management	1	3	2.1	3.8
070 String-Manipulation	1	4	2.7	.3
130 File I/O	1	5	3.4	.6
160 Application-Execution	1	1	.7	2.0
200 Message	1	7	4.8	4.3
210 Window-Creation	1	15	10.3	11.2
215 Multiple-Document Interface	1	13	8.9	1.9
...				
KEYWORDS (by Category and Hits)				
999 Dictionary entry not defined				
HMENU	1	1.4		
...				
050 Functionality does not exist				
GetProp	1	7.2		
...				
040 Logic changes required				
LoadIcon	1	2.5		
CREATESTRUCT	1	2.5		
WM_QUERYENDSESSION	1	2.0		
...				
030 Change with more / less API calls				
InvalidateRect	1	1.7		

WM_COMMAND	1	1.4
SetScrollPos	1	1.4
GetSubMenu	1	1.1
GetStockObject	1	1.1
...		
020 Replacement with parameter changes		
SendMessage	1	2.5
WPARAM	1	1.9
CW_USEDEFAULT	1	1.6
LPARAM	1	1.2
MessageBox	1	.8
HANDLE	1	.8
GetParent	1	.8
LRESULT	1	.6
WS_VSCROLL	1	.5
WS_HSCROLL	1	.5
GetWindow	1	.5
...		
010 Literal replacement		
CALLBACK	1	.0
LPSTR	1	.0
_MAX_PATH	1	.0
SB_VERT	1	.0
SB_HORZ	1	.0
RECT	1	.0
LOWORD	1	.0
HIWORD	1	.0
...		
000 Informational only		
NULL	1	.0
BOOL	1	.0
...		

Abb. 7.36: Ausführliche Aufstellung für reine OS/2-Portierung

Speziell die unter Kategorie 020 aufgeführten Elemente lassen sich oftmals mit Hilfe von UDMD und MiCL automatisch portieren. SMART liefert hier für einige Beispiele mit, unter anderem eines mit kompletter MDI-Unterstützung unter OS/2. In die nächste Analyse ziehen Sie diese Unterstützung mit ein.

1. Wählen Sie *Select Tables...* aus dem Menü *Tables*.

2. Im Dialog *Migration Tables* führen Sie die folgenden Schritte durch:

– Drücken Sie auf den Auswahlknopf *User UDMD with SMART Table*

– Selektieren Sie als *User UDMD Migration Data Base Directory* das Verzeichnis »smartaux\winxos2«.

– Drücken Sie auf *Set*, um diese Einstellungen abzuspeichern.

3. Wählen Sie *Analyze Source Code..* aus dem Menü *Analysis*.

4. Drücken Sie den Button *Process*, um die Analyse durchzuführen.

Wie der Abbildung entnehmen können, reduziert dies den Aufwand auf 85,6 Stunden bzw. 27,5 Prozent. Während dies nicht gerade berauschend ist, zeigt es auf, wie Sie schrittweise durch Erweiterung von UDMD, Schreiben von MiCL-Programmen oder Entwickeln zusätzlicher Kompatibilitätsschichten die direkte Portierung verbessern können.

SMART2 Source Migration - (E:\PRJ\BROWSER\ANALYSE\CODE.LST)

File Analysis Migration Resources Tables Options Help

Path/Filename	Size	Lines	Code	Hits	% Hits	Instances	% Inst.	Effort
E:\PRJ\BROWSER\								
BROWSER.C	36,350	759	562	341	60.7	146	26.0	85.6
BROWSER.H	985	26	16	0	.0	0	.0	.0
SubTotal (2)	37,335	785	578	341	59.0			85.6
Total (2)	37,335	785	578	341	59.0	146	25.3	85.6

Abb 7.37: SMART Analysebericht für reine OS/2-Portierung mit MDI-Unterstützung

Viel größere Ersparnis erreichen Sie aber mit Open32, wie die dritte Analyse beweist.

1. Wählen Sie *Select Tables...* aus dem Menü *Tables*.

2. Im Dialog *Migration Tables* führen Sie die folgenden Schritte durch:

– Selektieren Sie als *User UDMD Migration Data Base Directory* das Verzeichnis »smartaux\dapie32«.

– Drücken Sie auf *Set*, um diese Einstellungen abzuspeichern.

3. Wählen Sie *Analyze Source Code..* aus dem *Analysis*-Menü.

4. Drücken Sie den Button *Process*, um die Analyse durchzuführen.

SMART2 Source Migration - (E:\PRJ\BROWSER\ANALYSE\CODE.LST)

File Analysis Migration Resources Tables Options Help

Path/Filename	Size	Lines	Code	Hits	% Hits	Instances	% Inst.	Effort
E:\PRJ\BROWSER\								
BROWSER.C	36,350	759	562	369	65.7	149	26.5	3.8
BROWSER.H	985	26	16	0	.0	0	.0	.0
SubTotal (2)	37,335	785	578	369	63.8			3.8
Total (2)	37,335	785	578	369	63.8	149	25.8	3.8

Abb. 7.38: SMART-Analysebericht für Open32-Portierung

Die Abbildung zeigt, daß der Aufwand nahezu auf 0 reduziert ist. Eigentlich beträgt er exakt 0, jedoch enthält die bei SMART mitgelieferte Win32-nach-Open32-UDMD nicht alle mittlerweile in Open32 enthaltenen Elemente. In der ausführlichen Aufstellung sehen Sie, daß folgende Elemente nicht gefunden wurden:

– »MDITILE_VERTICAL«

– »MDITILE_HORIZONTAL«

– »WinMain«

– »WNDENUMPROC«

– »CHOOSEFONT«

– »min«

Ein entsprechendes Durchsuchen der Open32-Headerdateien »os2win.h«, »os2wdef.h« und »os2werr.h« ergibt, daß alle Elemente bis auf »WinMain« und »min« darin enthalten sind. Das Element »min« ist ein Makro aus der C-Standardbibliothek »stdlib.h«. Während es im Win32 SDK zusätzlich in der Datei »windef.h« definiert ist, ist es in Open32 nicht zusätzlich enthalten. Das Element »WinMain« ist eigentlich kein Win32-API-Aufruf, sondern die Einsprungsadresse eines Windows-Programms, die statt der auf allen anderen Plattformen üblichen C-Funktion »main« verwendet wird.

Ergänzen Sie die *Win32 nach Open32 UDMD* »Dapie32.dct« im SMART-Unterverzeichnis »SmartAux\Dapie32« mit den angegebenen Definitionen. Danach generieren Sie die UDMD neu, laden die Portierungstabellen nochmals und führen die gleiche Analyse nochmals durch.

```
.Keyword  = 030 000 130 MDITILE_VERTICAL
.SComment = MDITILE_VERTICAL is supported in OS/2
.LComment = MDITILE_VERTICAL is supported in OS/2
.Keyword  = 030 000 130 MDITILE_HORIZONTAL
.SComment = MDITILE_HORIZONTAL is supported in OS/2
.LComment = MDITILE_HORIZONTAL is supported in OS/2
.Keyword  = 040 000 130 WNDENUMPROC
.SComment = WNDENUMPROC is supported in OS/2
.LComment = WNDENUMPROC is supported in OS/2
.Keyword  = 040 000 130 CHOOSEFONT
.SComment = CHOOSEFONT is supported in OS/2
.LComment = CHOOSEFONT is supported in OS/2
.Keyword  = 010 000 130 WinMain
.SComment = WinMain is supported in OS/2
.LComment = WinMain is supported in OS/2
.Keyword  = 010 000 130 min
.SComment = min is supported in OS/2
.LComment = min is supported in OS/2
```

Abb. 7.39: Erweiterung der »dapie32.dct«

Wie Sie aus der nächsten Abbildung ersehen können, ist der Aufwand für Modifikationen an der Codebasis nun tatsächlich 0. Dies entspricht nun fast der Wahrheit, denn die Angaben für Include-Dateien stimmen noch nicht ganz.

Nach der Analyse löschen Sie die von SMART erzeugten temporären Dateien im Projekverzeichnis, indem Sie den Menüpunkt *Remove Temp. Analysis Files...* im Menü *Analysis* auswählen.

7.5.9 Analyse der Ressourcen-Dateien

Während Open32 viele Win32-Funktionen und -Nachrichten unter OS/2 bereitstellt, werden Windows-Ressourcen-Dateien und Ressourcen nicht direkt unterstützt. Dies ist nicht tragisch, da die Unterschiede der Windows- und OS/2-Implementierungen recht gering sind und eine automatische Konvertierung in der Regel möglich ist. Dies können Sie mit Hilfe der Analyse für Ressourcen-Dateien verifizieren.

Wie zuvor für die Codebasis erstellen Sie nun eine Listendatei »resource.lst«, die als einzige Datei »browser.rc« enthält. Die Analyse der Ressourcen-Dateien starten Sie wie folgt:

1. Wählen Sie *Analyze Resource Code...* im Menü *Analysis*.

2. Im Dialog *Resource Translation Analysis* führen Sie die folgenden Schritte durch:

– Selektieren Sie als *List-of-Resource-Files* die erstellte Datei »resource.lst«.

– Drücken Sie den Druckknopf *Process*, um die Analyse zu starten.

SMART zeigt das Ergebnis nicht im Fenster an, speichert es aber in der Datei »resource.sum« ab. Wie zu erwarten war, kann SMART die Ressourcen-Datei komplett umwandeln.

```
Resource Translation Analysis Report Summary

Total Resource File Count                           1
Total Resource File Size (Bytes)                1,312

Total Dialog Templates                              0
Total Dialog Code Lines                             0
Total Menu Template Code Lines                     30
Total StringTable Code Lines                        0
Total Accelerator Table Code Lines                  0

Estimated Translation Effort Without SMART         .2
Savings from SMART Resource Translator             .2

** Net Translation Effort **                       .0
Percent Effort Reduction using SMART            100.0 %
```

Abb. 7.40: Aufwandsersparnis für die Konvertierung der Ressourcen-Dateien

Nach der kompletten Analyse Ihrer realen Anwendung haben Sie eine Abschätzung über den zu leistenden Aufwand und letztendlich über die Kosten einer OS/2-Version. In unserem Beispiel spricht nichts gegen die Unterstützung der OS/2-Plattform.

Anpassung der gemeinsamen Codebasis

In den Quelltexten der gemeinsamen Codebasis müssen Sie gewährleisten, daß unter OS/2 die Include-Dateien für Open32 und unter Windows die für Windows eingebunden werden. Das folgende Listing zeigt die nötigen Änderungen der Includeliste in »browser.c«. Außerdem stolpert VisualAge for C++ über einen Typecast des linken Operanden in Zeile 750, den Sie einfach löschen können:

```
  (int) x = GetOpenFileName(&fn);
  ——-

#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#ifdef __OS2__
#include <os2win.h>
#else
#include <windows.h>
#include <commdlg.h>

#endif
#include "browser.h"
```

Abb. 7.41: Modifizierte Includeliste in »browser.c«

Bauen der OS/2-Anwendung

Wie Sie schon bei der Analyse festgestellt haben, enthält der Windows-Code »browser.c« als Einsprungsfunktion »WinMain« statt »main«. Sie finden im Verzeichnis der Open32-Beispiele des Developer's Toolkit ein Unterverzeichnis »WinMain«, in dem eine kleine Quelldatei »main.c« enthalten ist. Kopieren Sie aus dem Toolkit diese Datei »samples\open32\winmain\main.c« nach »\prj\browser\os2«. »main.c« enthält eine Einsprungsfunktion »main«, die die Funktion »WinMain« aufruft. Sie können »main.c« ohne Modifikationen benützen.

Außerdem benötigen Sie für die Erstellung eines OS/2-Programms eine sogenannte Moduldefinitionsdatei. Diese sollte bei Programmen, die das Open32-API benützen, eine Stapelgröße von mindestens 64 Kbyte definieren. Erstellen Sie die Datei »browser.def« mit folgendem Inhalt:

```
NAME Browser WINDOWAPI
DESCRIPTION "A simple file browser"
STACKSIZE 65536
```

Abb. 7.42: Die Moduldefinitionsdatei »browser.def«

Nun fehlt Ihnen noch eine Makedatei mit den Regeln, um Ihre OS/2-Anwendung zu bauen. Der Einfachheit halber verwenden Sie die bestehende Datei »browser.mak« aus dem Open32-Beispiel. Kopieren Sie die Datei »samples\open32\browser2\browser.mak« nach »\prj\browser\os2«. Die Makedatei bindet eine weitere Datei ein, die Regeln enthält, die von allen Beispielen aus dem Developer's Toolkit verwendet werden. Kopieren Sie diese Datei »samples\ibmsamp.inc« nach »\prj«.

Sie benützen die Windows Ressourcen-Dateien und Ressourcen als Original und erzeugen die OS/2-Versionen automatisch während des Baus der gesamten Anwendung. Dazu müssen Sie die angegebenen Regeln in »browser.mak« hinzufügen und in der Zeile

```
$(OBJDIR)\browser.res: $(HEADERS) browser.rc
```

die Datei »icon1.ico« in die Abhängigkeitsliste aufnehmen.

```
browser.rc: ..\browser.rc
   copy $** $@
   smartrc -O -R browser.rc

browser.h : ..\browser.h
   copy $** $@

browser.c : ..\browser.c
   copy $** $@

icon1.ico : ..\icon1.ico
   -smartcvt $** $@
```

Abb. 7.43: Zusatzregeln zum Kopieren und Konvertieren der gemeinsamen Quelldateien

Nun steht nichts mehr im Wege, Ihre erste Open32-Anwendung zu bauen. Führen Sie folgende Schritte durch:

1. Starten Sie ein OS/2-Eingabefenster

2. Wechseln Sie zu Ihrem OS/2-Projektverzeichnis:

```
cd \prj\browser\os2
```

3. Starten Sie den Bauprozeß:

```
nmake browser.mak
```

Sofern Sie alle Schritte korrekt befolgt haben, erhalten Sie keine Fehlermeldung, und die Datei »browser.exe« wird erzeugt. Die zahlreichen Warnungen während der Compilierung mit VisualAge for C++ sollten Sie nicht verunsichern. Der Compiler wird standardmäßig mit höchster Stufe für Warnungen eingestellt. Leider ist »browser.c« – wie zahlreiche Windows-programme –sehr unsauber programmiert. Sollten Sie mit VisualAge for C++ Unsauberkeiten in Ihrem Quellcode entdecken, sollten Sie diese am besten beseitigen.

Starten Sie Ihre Open32-Anwendung »Browser«, mit der Sie mehrere Dateien gleichzeitig öffnen und lesen können.

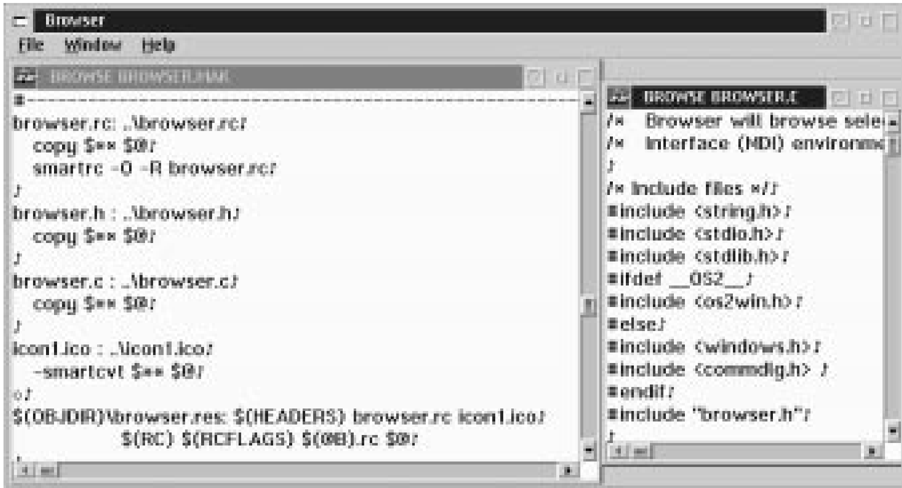


Abb. 7.44: Die Open32-Anwendung »Browser«

Falls Ihre Anwendung sich nicht auf das Open32-API beschränkt, können Sie verschiedene Techniken einsetzen, um mit Ihrer bestehenden Windows-Anwendung beide Plattformen zu unterstützen:

– Separieren Sie den gemeinsamen Code

– Entwickeln Sie den plattformspezifischen Code doppelt

– Portieren Sie den plattformspezifischen Code mit Hilfe von SMART

– Entwickeln Sie eine dünne Portabilitätsschicht unter OS/2, die den entsprechenden Windows-Funktionen entspricht und diese mit Hilfe entsprechender OS/2-Funktionen implementiert. Ein guter Kandidat sind zum Beispiel »Named Pipes«. Ihr bisheriger Quellcode bleibt unverändert.

– Entwickeln Sie Fensterklassen für Windows-spezifische Controls (zum Beispiel das Tree-View-Control) mit Hilfe vergleichbarer OS/2-Controls. Ihr bisheriger Quellcode bleibt unverändert.

Entnehmen Sie der SMART-Online-Dokumentation, wie Sie Windows-Code mit Hilfe von SMART auf OS/2 portieren können, Die weiteren Techniken finden Sie in [10] näher erläutert.

Nähere Informationen zu den angesprochenen Produkten, Komponenten oder Services finden Sie im Internet:

IBM Redbooks	www.redbooks.ibm.com
Solutions Provider Marketing (SPM)	www.developer.ibm.com
IBM Application Development	www.software.ibm.com/ad
VisualAge for C++	www.software.ibm.com/ad/visualage

Falls Sie als Windows-Entwickler Interesse an der Portierung auf OS/2 haben, sollten Sie hier weiterlesen:

1. Bücher:

[1] VisualAge for C++ for Windows – Programming Guide, Chapter »Porting Programs from VisualAge for C++ for OS/2«. IBM, 1996. S33H-5032-00

[2] (FAQ) Portable GUI Development Kits. Newsgroup: comp.windows.misc

[3] Solutions for Common Source and Multiple Platform Development – Overview. IBM White Paper, May 1995

[4] IBM Develper API Extensions for OS/2 and Source Migration Analysis Reporting Toolset. IBM White Paper, May 1995

[5] Open32 Programming Guide and Reference. IBM Developer's Toolkit for OS/2 Warp, Version 4.0. IBM, 1996

[6] The SMART Programming Guide for Portability, Version 2.1B. SMART2 Toolset. One Up Corporation, 1995

[7] Anthony S. Glad: Cross-Platform Software Development. Van Nostrand Reinhold, 1995. ISBN 0-442-01812-6

[8] Mark R. Horton: Portable C Software. Prentice-Hall, 1990. ISBN 0-138-68050-7

[9] Microsoft: Windows-Dokumentation

[10] Mike Foster et al: Open32 for OS/2 Warp. IBM International Technical Support Organization, 1996

2. Zeitschriftenartikel zum Thema

Harald Wilhelm: Wechselbalg. OS/2 Inside, 9/96, S. 53f. AWi Verlag, 1996

Derrel Blain: Porting Applications to OS/2: A Historical Review of OS/2. OS/2 Developer, March/April 1995, S. 49ff. Miller Freeman, 1995

8 Die DOS- und Windows-Umgebung

von Uwe Schlenther

Die Unterstützung für DOS- und Windows-Programme ist eine der wichtigsten Komponenten von OS/2. Bereits bei der Planung des neuen Betriebssystems wurde berücksichtigt, daß niemand gezwungen sein soll, die bereits gekauften und eingerichteten Anwendungsprogramme für das DOS-Betriebssystem beim Umstieg auf OS/2 über Bord zu werfen.

Bei den ersten Versionen von OS/2 wurde das Ziel einer brauchbaren DOS-Emulation nicht erreicht. DOS-Programme konnten nur mit vielen Einschränkungen ablaufen, weshalb die DOS-Umgebung oft scherzhaft als »Penalty Box« bezeichnet wurde. Beispielsweise konnte immer nur ein DOS-Programm gleichzeitig verwendet werden, das zudem nur als Vordergrundanwendung ablaufen konnte. Im wesentlichen waren diese Probleme durch Mängel in den damals verwendeten 16 Bit-Prozessoren verursacht.

Mit der Einführung des Prozessors 80386 schuf Intel die Voraussetzung für eine wesentlich verbesserte DOS-Umgebung in OS/2. Es wurde eine neue Prozessor-Betriebsart, der »Virtual 8086 Mode«, eingeführt. In dieser Betriebsart kann das Betriebssystem ein DOS-Programm in einem geschützten Bereich, einer sogenannten virtuellen DOS-Maschine (VDM), ablaufen lassen. Dadurch bekommt das Programm das Gefühl, einen vollständigen Rechner für sich alleine zu haben. Das Betriebssystem behält dabei die vollständige Kontrolle über das ablaufende Programm. Auf diese Weise wurde es möglich, mehrere DOS-Programme gleichzeitig und vollständig abgeschirmt voneinander und vom Betriebssystem ablaufen zu lassen.

Die DOS-Umgebung von OS/2 2.0, der ersten-32-Bit-Version des Betriebssystems, funktionierte bereits hervorragend. Sie stand einem gewöhnlichen DOS-System in nichts nach und übertraf die zu dieser Zeit verfügbaren DOS-Systeme in vielen Bereichen. Durch eine speziell angepaßte Version von Windows 3.0 war es auch möglich, für diese Umgebung geschriebene Programme unter OS/2 ablaufen zu lassen. Beide Umgebungen wurden beim Update auf OS/2 2.1 nochmals weiterentwickelt und vervollständigt. Nun konnten auch Windows-3.1-Programme verwendet werden, sowie viele derjenigen DOS-Programme, die unter OS/2 2.0 noch Schwierigkeiten bereiteten.

Seit OS/2 Warp Version 3 kann man die DOS-Umgebung von OS/2 als die perfekteste DOS-Umgebung überhaupt bezeichnen. Es gibt nur noch vereinzelt DOS-Programme, die nicht unter OS/2 funktionieren, und die meisten DOS-Anwendungen laufen unter OS/2 besser und schneller als unter echtem DOS oder in der DOS-Umgebung eines anderen Betriebssystems. Die wesentlichen Vorteile der DOS-Emulation von OS/2 gegenüber dem meisten anderen DOS-Umgebungen sind:

– Ein 32-Bit-Betriebssystem als Basis sorgt für hohe Ablaufgeschwindigkeit.

– Multitasking mit mehreren DOS-, Windows- und OS/2-Programmen ist möglich.

– Jedes einzelne DOS-Programm läuft in einem eigenen Adreßraum ab. Dadurch kann ein DOS-Programm – auch im Falle eines Absturzes – nicht andere Programme oder das Betriebssystem beeinträchtigen.

– Jedem DOS-Programm steht praktisch der gesamte DOS-Speicher zur Verfügung. Treiber wie beispielsweise für Maus oder CD-ROM müssen nicht in der DOS-Umgebung geladen werden, da entsprechende Schnittstellen von OS/2 zur Verfügung gestellt werden.

– Jede DOS-Sitzung läßt sich individuell konfigurieren, also zum Beispiel mit EMS- oder XMS-Speicher ausstatten. DOS-Sitzungen mit verschiedenen Einstellungen können parallel ablaufen. Werden die Einstellungen einer DOS-Sitzung geändert, werden diese Änderungen *ohne* Neustart des Rechners wirksam.

– DOS-Anwendungen können mehr Speicher benutzen, als im Rechner tatsächlich vorhanden ist. Dieser Speicher wird in Form *virtuellen* Speichers von OS/2 zur Verfügung gestellt.

Durch die hervorragende Unterstützung von DOS- und Windows-Programmen bekommt der OS/2-Benutzer neben dem großen Angebot an OS/2-Software auch Zugang zu einer der größten Software-Bibliotheken der Welt. Damit ist OS/2 das Betriebssystem mit der reichhaltigsten Software-Auswahl überhaupt.

DOS- und Windows-Programme können unter OS/2 genauso behandelt werden wie OS/2-Programme. Ein Doppelklick auf das Programmobjekt oder auf das Dateiobjekt der Exe- oder Com-Datei startet automatisch die richtige Umgebung (DOS oder Windows) und das Programm wird ausgeführt. Genauso ist es möglich, auf der Befehlszeile den Namen der Exe- oder Com-Datei in einer DOS- oder OS/2-Befehszeile einzugeben und die Eingabetaste zu drücken.

Anders als ein OS/2-Programm ist ein DOS- oder Windows-Programm aber nicht in der Lage, selbst für optimale Versorgung mit Systemressourcen zu sorgen. Ein DOS- oder Windows-Programm »weiß« ja nicht, daß es unter OS/2 läuft. Aus diesem Grund muß der Benutzer selbst dafür sorgen, daß das DOS- oder Windows-Programm optimal konfiguriert wird. Glücklicherweise verfügt OS/2 über genügend Einstellungsmöglichkeiten, um den meisten Programmen bestmögliche Arbeitsbedingungen gewährleisten zu können. Solche Einstellungen sind immer nur für dasjenige Programmobjekt gültig, bei dem die sonst gültigen Standardeinstellungen verändert wurden. Auf diese Weise können die Einstellungen für jedes Objekt separat verändert werden und es können mehrere DOS-Umgebungen oder DOS-Programme mit verschiedenen Einstellungen zur selben Zeit aktiv sein.

8.1 Automatische Konfiguration von DOS- und Windows-Programmen

Damit auch ungeübte Benutzer sofort von der Konfigurierbarkeit von DOS- und Windows-Anwendungen profitieren können, kann OS/2 gängige DOS- und Windows-Programme automatisch optimal konfigurieren. Hierzu verfügt OS/2 über eine Datenbank, in der zu über 1000 Anwendungen die optimalen Einstellungen verzeichnet sind. Diese »Umstellungs-Datenbank« kommt normalerweise bereits bei der Installation von OS/2 zum Einsatz.

Kurz vor Ende des Installationsvorgangs fragt das Installationsprogramm, ob die Festplatte nach vorhandenen Anwendungen durchsucht werden soll. Für jede Anwendung, die auf der Festplatte vorhanden und in der Umstellungsdatenbank verzeichnet ist, wird dann automatisch ein Programm-Objekt auf der Arbeitsoberfläche erzeugt. Hinter dieser automatischen Funktion verbirgt sich das Programm »Programme hinzufügen«, das Sie im Ordner OS/2 System – Systemkonfiguration vorfinden. Sie können dieses Programm jederzeit wieder starten, um Programmobjekte für neu hinzugekommenen Anwendungen erzeugen zu lassen.

Die automatische Konfiguration wird auch beim manuellen Erzeugen eines Programmobjekts aktiv. Sobald Sie den Namen der Programmdatei im Einstellungsnotizbuch eingetragen haben, durchsucht OS/2 die Umstellungsdatenbank nach diesem Programm. Wird eine passende Anwendung gefunden, werden die dazu gespeicherten Einstellungen automatisch in dieses Programmobjekt übernommen. An der Symbol-Seite im Einstellungsnotizbuch kann man erkennen, ob das fragliche Programm in der Datenbank gefunden wurde: Steht dort bereits der korrekte Name der Anwendung, dann waren Einstellungen in der Datenbank vorhanden. Andernfalls steht dort der Standardwert »Programm«.

8.2 Manuelle Konfiguration von DOS- und Windows-Programmen

Ist eine Anwendung nicht in der Umstellungs-Datenbank enthalten, ist das noch immer kein Malheur. Die meisten DOS- und Windows-Programme laufen völlig problemlos mit den Standardeinstellungen für den jeweiligen Sitzungstyp. Nur wenn Probleme mit einer Anwendung auftreten oder wenn man einer Anwendung etwas mehr Arbeitsspeicher gönnen möchte, müssen die Einstellungen für die betreffende Sitzung geändert werden.

Um die Einstellungen der DOS- oder Windows-Umgebung zu verändern, öffnen Sie das Kontextmenü eines DOS- oder Windows-Programmobjekts und wählen Sie *Einstellungen*. Es erscheint das zugehörige Notizbuch, in dem Sie die Sitzung-Seite auswählen

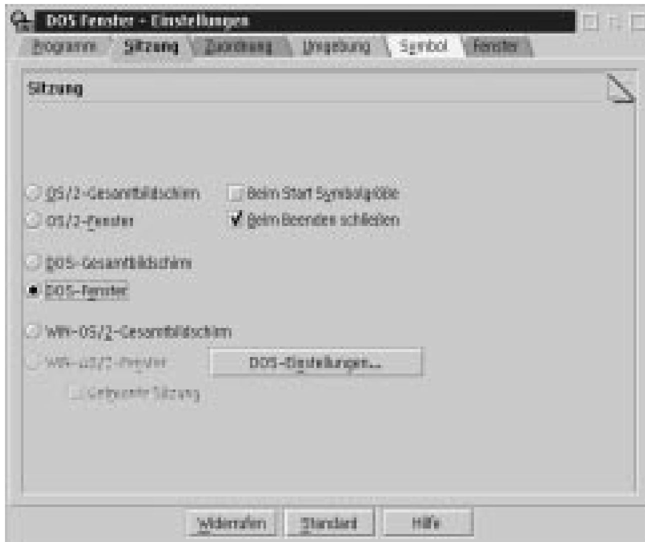


Abb. 8.1:
Das *Einstellungen*-Notizbuch eines Programmobjekts

Auf dieser Seite wird zunächst ausgewählt, welche Umgebung für das betreffende Programm gestartet werden soll. Normalerweise wird diese Einstellung von OS/2 vorgenommen. Der Benutzer muß lediglich auswählen, ob das Programm in einem Fenster auf der OS/2-Arbeitsoberfläche oder, wie bei DOS-Programmen üblich, im Gesamtbildschirm-Modus gestartet werden soll. DOS-Programme, die Grafik verwenden, sollten immer im Gesamtbildschirm-Modus ausgeführt werden, da viele Grafiktreiber die Ausführung grafischer Programme im Fenster nicht unterstützen. Die Ausführung von Windows-Anwendungen im Fenster (»Seamless Windows«) wird mittlerweile von allen Grafiktreibern unterstützt und ist, trotz des etwas höheren Speicherverbrauchs und der etwas geringeren Geschwindigkeit der Ausführung im Gesamtbildschirm-Modus, vorzuziehen. Einige Windows-Programme wollen aber unbedingt im Gesamtbildschirm-Modus ausgeführt werden. Falls einmal ein Windows-Programm im Seamless-Mode Schwierigkeiten bereiten sollte, so versuchen Sie, es im Gesamtbildschirm-Modus ablaufen zu lassen.

Der weitaus wichtigste Teil der Einstellungsmöglichkeiten für DOS- und Windows-Programme verbirgt sich hinter dem Button DOS-Einstellungen bzw. Win-OS/2-Einstellungen. Wird dieser Knopf betätigt, erscheint zunächst ein Dialogfenster, in dem ausgewählt werden kann, welche Gruppe von Einstellungen bearbeitet werden soll. Erst dann können die Einstellungen für das Programm bearbeitet werden. Im nächsten Abschnitt dieses Kapitels wird jede einzelne dieser Einstellungsmöglichkeiten für DOS- und Windows-Programme ausführlich erklärt.

Auf der Seite *Sitzung* im Notizbuch *Einstellungen* von Programmobjekten befindet sich noch eine Einstellung, die nur für Windows-Fenstersitzungen verfügbar ist: »Getrennte Sitzung«. Sollen mehrere Windows-Anwendungen unter OS/2 gestartet werden, wird normalerweise nur für die erste gestartete Anwendung die Windows-Umgebung erzeugt. Die weiteren Windows-Anwendungen verwenden dann die Ressourcen dieser Umgebung gemeinsam mit der zuerst

gestarteten Windows-Anwendung. Stürzt nun eins der Windows-Programme ab, kann dabei auch die Windows-Umgebung zerstört werden. Die anderen in dieser Umgebung ablaufenden Windows-Programme müssen dann ebenfalls beendet werden. Ist bei einem Programm-Objekt die Einstellung »Getrennte Sitzung« aktiviert, wird für *dieses* Programm eine Windows-Umgebung erzeugt, unabhängig davon, ob bereits eine andere Windows-Umgebung gestartet wurde. Stürzt das Programm nun ab, wird davon nur seine eigene Windows-Umgebung und nicht gleichzeitig ablaufende Windows-Anwendungen beeinflußt. Da eine Windows-Umgebung sehr viel Arbeitsspeicher benötigt, sollte man die »Getrennte Sitzung« nur dann aktivieren, wenn sich ein Windows-Programm als besonders problematisch erweist.

8.3 Die DOS- und Windows-Einstellungen im einzelnen

Die Einstellungen für DOS- und Windows-Programme sind in verschiedene Gruppen aufgeteilt. Ferner deutet ein Kürzel am Beginn der Bezeichnung einer Einstellung auf den beeinflußten Bereich der Umgebung hin. Es gibt folgende Gruppen von Einstellungen:

– DOS-Tastatureinstellungen

– DOS-Hauptspeichereinstellungen

– DOS-Einstellungen für Maus und Tastbildschirm

– DOS-Druckereinstellungen

– DOS-Bildschirmeinstellungen

– Andere DOS-Einstellungen

– Win-OS/2-Einstellungen

Die Win-OS/2-Einstellungen stehen nur bei Windows-Programmen zur Verfügung. Die anderen Einstellungen sind sowohl bei DOS- als auch bei Windows-Anwendungen verfügbar.

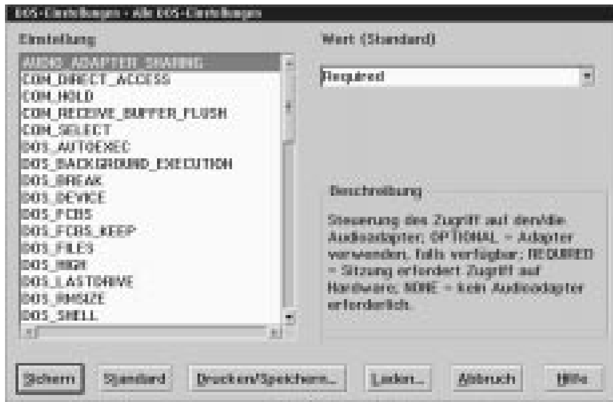


Abb. 8.2: Der DOS-Einstellungsdialog

Im Dialogenster für DOS- und Win-OS/2-Einstellungen (siehe Abb. 8.2) findet man Buttons zum Sichern der Änderungen und Wiederherstellen der Standardeinstellung für die ausgewählte Option. Außerdem gibt es Knöpfe zum Ausdrucken und Abspeichern der Einstellungen und zum Laden von Einstellungen aus einer Datei. Letztere Funktion ist gut geeignet, um einen Satz von Einstellungen von einem anderen OS/2-System zu übernehmen, wenn eine bestimmte DOS- oder Windows-Anwendung dort mit optimalen Einstellungen versehen wurde. Die Funktionen zum Abspeichern und Ausdrucken der Einstellungen sind zu einer Funktion zusammengefaßt, die Auswahl des Ausgabeziels erfolgt erst nach Betätigung des entsprechenden Knopfes.

8.3.1 DOS-Tastatureinstellungen

Die DOS-Tastatureinstellungen beginnen mit dem Kürzel *KBD* (*Keyboard*). Folgende Einstellungen stehen hier zur Verfügung:

KBD_ALTHOME_BYPASS

Die Tastenkombination [Alt]-[Pos1] dient unter OS/2 normalerweise zum Umschalten zwischen Fenster- und Gesamtbildschirmdarstellung von DOS-Sitzungen. Falls eine DOS-Anwendung diese Tastenkombination zum Auslösen einer Funktion benutzt, sollte diese Einstellung auf *On* geschaltet werden. Es ist dann allerdings nicht mehr möglich, mit [Alt]-[Pos1] zwischen Fenster und Gesamtbildschirm umzuschalten. Ein Menüpunkt zum Umschalten zwischen beiden Betriebsarten befindet sich jedoch stets im Kontextmenü des Programmobjekts einer ablaufenden DOS-Anwendung. Die Voreinstellung für *KBD_ALTHOME_BYPASS* ist *Off*.

KBD_BUFFER_EXTEND

Wird eine Taste betätigt, wird dieses Ereignis in einem begrenzten Pufferbereich gespeichert. Wenn eine Anwendung die anstehenden Tastaturereignisse nicht schnell genug bearbeitet, ertönt ein Piepston, sobald im Puffer kein Platz mehr verfügbar ist. OS/2 kann diesen Puffer

für jede DOS-Sitzung vergrößern, wenn die Einstellung *KBD_BUFFER_EXTEND* auf *On* steht. Diese Vergrößerung des Puffers geht zu Lasten des Arbeitsspeichers. Sie können einige Bytes Arbeitsspeicher freihalten, wenn Sie diese Einstellung auf *Off* ändern. Die Voreinstellung ist *On*.

KBD_CTRL_BYPASS

Die Tastenkombinationen [Strg]-[Esc] und [Alt]-[Esc] haben unter OS/2 die Funktion, zur Fensterliste beziehungsweise zum nächsten geöffneten Objekt umzuschalten. Falls eine DOS-Anwendung eine dieser Tastenkombinationen zum Auslösen einer eigenen Funktion benutzt, sollten Sie diese Kombination hier für die Verwendung innerhalb der DOS-Sitzung freischalten. Die gewählte Tastenkombination wird dann innerhalb dieser DOS-Sitzung vom OS/2-System nicht beachtet. Die Voreinstellung ist *NONE*, das heißt, keine der beiden Tastenkombinationen wird an die DOS-Sitzung weitergeleitet.

KBD_RATE_LOCK

Bleibt eine Taste auf der Tastatur gedrückt, wird das betreffende Zeichen nach einer kurzen Pause mit einer bestimmten Geschwindigkeit wiederholt ausgelöst. Diese Geschwindigkeit kann normalerweise von Anwendungen beeinflußt werden. Möchten Sie die Änderung der Wiederholungsgeschwindigkeit verhindern, wählen Sie bei dieser Einstellung *On*. Die Voreinstellung ist *Off*, Anwendungen können also den Geschwindigkeitswert verändern.

8.3.2 DOS-Hauptspeichereinstellungen

Unter den DOS-Hauptspeichereinstellungen befinden sich neben den Einstellungen, die mit den Kürzeln *DPMI*, *EMS*, *MEM* und *XMS* beginnen, auch einige Einstellungen mit dem Kürzel *DOS*. Im einzelnen sind es folgende Einstellungsmöglichkeiten:

DOS_HIGH

Wird diese Einstellung auf *On* gesetzt, wird der DOS-Kern für die betreffende Sitzung in einen Speicherbereich oberhalb von 1 Mbyte geladen. Auf diese Weise bleibt mehr konventioneller DOS-Speicher für Anwendungen verfügbar. Die Voreinstellung ist *Off* und sollte für die meisten DOS-Anwendungen ausreichen. Bei Anwendungen, die besonders viel konventionellen Arbeitsspeicher benötigen, sollte hier *On* eingestellt werden. Um die Einstellung *On* für alle DOS-Sitzungen zu aktivieren, ändern Sie in der »config.sys« von OS/2 die Zeile

```
DOS=LOW,NOUMB
```

in

```
DOS=HIGH,NOUMB
```

DOS_RMSIZE

Mit dieser Einstellung geben Sie an, wieviel konventioneller DOS-Speicher in der betreffenden Sitzung verfügbar sein soll. Es können Werte zwischen 128 und 640 Kbyte in Schritten von 4 Kbyte eingestellt werden. Die Änderung auf einen Wert unterhalb von 640 Kbyte ist dann sinnvoll, wenn dem OS/2-System nur sehr wenig Arbeitsspeicher zur Verfügung steht

oder wenn der minimale Speicherbedarf eines DOS-Programms ermittelt werden soll. Soll global ein anderer Wert für *DOS_RMSIZE* verwendet werden, so ändern Sie in der »config.sys« von OS/2 die Zeile

```
RMSIZE=640
```

entsprechend ab.

DOS_UMB

Neben dem konventionellen Arbeitsspeicher verfügt DOS über die sogenannten »Upper Memory Blocks«. Dieser Speicherbereich kann entweder von einer Anwendung oder von DOS verwaltet werden. Um die Verwaltung durch DOS zu aktivieren, schalten Sie diese Einstellung auf *On*. Es ist dann möglich, Treiber und speicherresidente Programme mit dem Kommando LOADHIGH in diesen Speicherbereich zu laden, um den konventionellen Arbeitsspeicher zu entlasten. Bei der Voreinstellung *Off* muß die Verwaltung der Upper Memory Blocks durch eine Anwendung erfolgen, ansonsten bleiben sie ungenutzt. Um die Einstellung *On* für alle DOS-Sitzungen zu aktivieren, ändern Sie in der »config.sys« von OS/2 die Zeile

```
DOS=LOW,NOUMB
```

in

```
DOS=LOW,UMB
```

DPMI_DOS_API

Diese Einstellung kontrolliert die Umsetzung von DPMI-Aufrufen innerhalb der DOS-Sitzung. DPMI (*DOS Protected Mode Interface*) ist eine Möglichkeit, unter DOS den Protected Mode des Prozessors zu benutzen, um auf Speicherbereiche oberhalb von 1 Mbyte zuzugreifen. Die meisten DPMI-fähigen DOS-Programme kommen mit der Voreinstellung *AUTO* zurecht. Sollte eine solche Anwendung Schwierigkeiten bereiten, versuchen Sie, hier *ENABLED* einzustellen. Falls eine gewöhnliche DOS-Anwendung Probleme mit der DPMI-Umsetzung hat, stellen Sie hier *DISABLED* ein.

DPMI_MEMORY_LIMIT

Geben Sie hier an, wieviel Speicher einer DPMI-fähigen Anwendung zur Verfügung gestellt werden soll. Der Maximalwert liegt bei 512 Mbyte, die Voreinstellung ist 16 Mbyte für DOS- und 64 Mbyte für Win-OS/2-Sitzungen. Die Einstellung kann in Schritten von 1 Mbyte verändert werden. Wird ein Wert von 0 Mbyte eingestellt, wird die Unterstützung für DPMI in dieser Sitzung deaktiviert.

Wie alle Speichereinstellungen kann auch diese unabhängig vom tatsächlich installierten Arbeitsspeicher verändert werden. Fordert eine Anwendung mehr Speicher an, als im Rechner vorhanden ist, wird er von OS/2 simuliert. Dies wird durch Auslagerung nicht benutzter Speicherbereiche auf die Festplatte des Rechners realisiert. Außerdem wird angeforderter Speicherplatz erst dann tatsächlich belegt, wenn er von der Anwendung zum Speichern von Daten benutzt wird. Die Speichereinstellungen können also im Einzelfall ruhig großzügig bemessen werden.

DPMI_NETWORK_BUFF_SIZE

Stellen Sie hier die Größe des Netzwerk-Umsetzungspuffers für in dieser Sitzung ablaufende Anwendungen ein. Der Wert kann von 1 bis 64 Kbyte in Schritten von 1 Kbyte beeinflußt werden. Die Voreinstellung ist 8 Kbyte. Diese Einstellung sollte nur dann verändert werden, wenn die Dokumentation der verwendeten Anwendung dies verlangt.

EMS_FRAME_LOCATION

OS/2 kann DOS-Anwendungen Speicher nach der Expanded Memory Specification (LIM-EMS 4.0) zur Verfügung stellen. Dazu wird ein Speicherbereich von 64 Kbyte unterhalb der 1 Mbyte-Grenze in den Bereiche des Speichers oberhalb 1 Mbyte eingeblendet. Die Position dieses Speicherfensters wird normalerweise automatisch festgelegt (Voreinstellung *AUTO*). Sollte dies zu einem Konflikt mit einem Gerätetreiber führen, wählen Sie aus der Liste der Einstellungsmöglichkeiten eine andere Startadresse für den Pufferbereich aus.

EMS_HIGH_OS_MAP_REGION

Hier können Sie die Größe eines weiteren EMS-Bereichs angeben. Der Wert kann von 0 Kbyte bis 96 Kbyte in Schritten von 16 Kbyte beeinflußt werden. Voreinstellung ist 0 Kbyte.

Mit Hilfe dieser Einstellung und den Einstellungen *MEM_EXCLUDE_REGIONS* sowie *MEM_INCLUDE_REGIONS* können Adressenkonflikte verhindert werden, wenn Geräte verwendet werden, für die kein virtueller Gerätetreiber für die DOS-Umgebung verfügbar ist.

EMS_LOW_OS_MAP_REGION

Ist eine DOS-Anwendung in der Lage, konventionellen DOS-Speicher als EMS-Speicher zu benutzen, kann hier dessen Größe eingestellt werden. Der Wert kann zwischen 0 Kbyte und 576 Kbyte in Schritten von 16 Kbyte verändert werden. Voreingestellt sind 384 Kbyte.

EMS_MEMORY_LIMIT

Stellen Sie hier ein, wieviel EMS-Speicher in der DOS-Sitzung verfügbar sein soll. Sie können Werte zwischen 0 Kbyte und 32768 Kbyte (32 Mbyte) in Schritten von 16 Kbyte einstellen. Voreingestellt ist ein Wert von 2048 Kbyte (2 Mbyte).

Wie alle Speichereinstellungen kann auch diese unabhängig vom tatsächlich installierten Arbeitsspeicher verändert werden. Fordert eine Anwendung mehr Speicher an, als im Rechner vorhanden ist, wird der Speicher von OS/2 simuliert. Dies wird durch Auslagerung nicht benutzter Speicherbereiche auf die Festplatte des Rechners realisiert. Außerdem wird angeforderter Speicherplatz erst dann tatsächlich belegt, wenn er von der Anwendung zum Speichern von Daten benutzt wird. Die Speichereinstellungen können also auch hier im Einzelfall ruhig großzügig bemessen werden.

MEM_EXCLUDE_REGIONS

Unter gewissen Umständen müssen Speicherbereiche zwischen 640 Kbyte und 1 Mbyte vor der Verwendung durch Anwendungen geschützt werden. Dies ist beispielsweise dann der Fall, wenn eine Erweiterungskarte einen solchen Speicherbereich zum Datentransfer benutzt. Die Einstellung *MEM_EXCLUDE_REGIONS* erfüllt genau diese Aufgabe. Sie können wahlweise

die Startadresse eines 4 Kbyte großen Bereichs (z.B. D0000) oder einen Bereich beliebiger Größe (z.B. C0000-C4000) in hexadezimaler Schreibweise angeben. Man kann auch beide Möglichkeiten kombinieren (z.B. D0000,C0000-C4000). Normalerweise ist das Eingabefeld leer.

MEM_INCLUDE_REGIONS

Ist durch die spezielle Konfiguration Ihres Rechners zusätzlicher Speicher zwischen 640 Kbyte und 1 Mbyte für die Verwendung durch Anwendungsprogramme verfügbar, können Sie diese Speicherbereiche unter der Einstellung *MEM_INCLUDE_REGIONS* angeben und dadurch nutzbar machen. Sie können wahlweise die Startadresse eines 4 Kbyte großen Bereichs (z.B. D0000) oder einen Bereich beliebiger Größe (z.B. C0000-C4000) in hexadezimaler Schreibweise angeben. Man kann auch beide Möglichkeiten kombinieren (z.B. D0000,C0000-C4000). Normalerweise ist dieses Eingabefeld leer.

XMS_HANDLES

OS/2 kann DOS-Anwendungen zusätzlichen Speicher nach der Extended Memory Specification (XMS) zur Verfügung stellen. Die Zuweisung dieses Speicher erfolgt über spezielle Verwaltungsstrukturen, sogenannte *Handles*. Hier können Sie einstellen, wieviele dieser Handles in der DOS-Sitzung vorhanden sein sollen. Beachten Sie, daß jedes zusätzliche XMS-Handle eine gewisse Menge Speicher belegt. Sie sollten also nicht mehr Handles vergeben als unbedingt notwendig. Die Anzahl der Handles kann im Bereich von 0 bis 128 eingestellt werden. Die Voreinstellung liegt bei 32 Handles. Dieser Wert ist für die meisten Anwendungen ausreichend.

XMS_MEMORY_LIMIT

Stellen Sie hier ein, wieviel Speicher nach der Extended Memory Specification (XMS) innerhalb der betreffenden DOS-Umgebung verfügbar sein soll. Der Wert kann von 0 Kbyte bis zu 16384 Kbyte (16 Mbyte) in Schritten von 4 Kbyte verändert werden. Die Voreinstellung ist 2048 Kbyte (2 Mbyte).

Anders als bei DOS/Windows verwendet Win-OS/2 keinen XMS-Speicher sondern DPMI-Speicher. Um einer Windows-Anwendung mehr Speicher zur Verfügung zu stellen, sollte also die Einstellung *DPMI_MEMORY_LIMIT* verwendet werden.

Wie alle Speichereinstellungen kann auch diese unabhängig vom tatsächlich installierten Arbeitsspeicher verändert werden. Fordert eine Anwendung mehr Speicher an, als im Rechner vorhanden ist, so wird er von OS/2 simuliert. Dies wird durch Auslagerung nicht benutzter Speicherbereiche auf die Festplatte des Rechners realisiert. Außerdem wird angeforderter Speicherplatz erst dann tatsächlich belegt, wenn er von der Anwendung zum Speichern von Daten benutzt wird. Die Speichereinstellungen können also auch hier im Einzelfall ruhig großzügig bemessen werden.

XMS_MINIMUM_HMA

Durch eine Besonderheit bei der XMS-Speicherverwaltung kann ein knapp 64 Kbyte großer Speicherbereich oberhalb von 1 Mbyte, die sogenannte High Memory Area (HMA), mit Einschränkungen als konventioneller Arbeitsspeicher genutzt werden. Typischerweise wird dieser

Speicherbereich von speicherresidenten Programmen (TSR) verwendet, wodurch der Speicherbereich unter der 640 Kbyte-Grenze entlastet wird. Leider kann jeweils nur ein Programm die HMA verwenden. Es ist daher sinnvoll, ein möglichst großes TSR-Programm in die HMA zu laden. Mit der Einstellung *XMS_MINIMUM_HMA* kann angegeben werden, wie groß ein TSR-Programm mindestens sein muß, damit ihm Zugang zur HMA gewährt wird.

Der Wert für *XMS_MINIMUM_HMA* kann von 0 Kbyte bis 63 Kbyte in Schritten von 1 Kbyte beeinflußt werden. Die Voreinstellung liegt bei 0 Kbyte, d.h. das erstbeste TSR-Programm darf die HMA verwenden.

8.3.3 DOS-Einstellungen für Maus und Tastbildschirm

In dieser Gruppe findet sich momentan nur eine Einstellungsmöglichkeit:

MOUSE_EXCLUSIVE_ACCESS

Läuft eine DOS-Anwendung im Fenster auf der OS/2-Arbeitsoberfläche ab, wird normalerweise der Mauszeiger von OS/2 zur Steuerung benutzt. Falls ein DOS-Programm selbst Mausunterstützung bereitstellt, kann unter Umständen ein zweiter Mauszeiger im DOS-Fenster sichtbar sein. Tritt bei einer Anwendung dieses Problem auf, sollten Sie *MOUSE_EXCLUSIVE_ACCESS* auf *On* stellen. Die Voreinstellung lautet *Off*.

Falls am Rechner ein Tastbildschirm mit einem entsprechenden Gerätetreiber angeschlossen ist, gibt es eine weitere Einstellungsmöglichkeit namens *TOUCH_EXCLUSIVE_ACCESS*, die der Einstellung *MOUSE_EXCLUSIVE_ACCESS* entspricht: Stellt man sie auf *On*, wird eine Berührung des Bildschirms in eine Aktion innerhalb des Anwendungsfensters umgesetzt. Bleibt die Voreinstellung *Off* ausgewählt, reagiert die Anwendung nur auf Berührungen, die im Anwendungsfenster selbst stattfinden.

8.3.4 DOS-Druckereinstellungen

Die DOS-Druckereinstellungen beginnen mit dem Kürzel *PRINT*.

PRINT_SEPARATE_OUTPUT

OS/2 verfügt über ein relativ aufwendiges Drucksystem. Das ist notwendig, da OS/2 auch im Multitaskingbetrieb die Druckausgabe verschiedener Anwendungen auseinanderhalten können muß. Während OS/2-Programme dem Drucksystem selbst mitteilen können, wann ein Druckauftrag abgeschlossen ist, muß bei DOS-Anwendungen auf andere Weise festgestellt werden, wann ein Druckauftrag beendet ist.

Normalerweise betrachtet OS/2 das Beenden eines DOS-Programms als Ende der Druckausgabe. Manche Anwendungen rufen allerdings weitere Programme auf, die zur Druckausgabe beitragen. In diesem Fall sollte die Einstellung *PRINT_SEPARATE_OUTPUT* auf *Off* geschaltet werden. Beläßt man die Voreinstellung *On*, wird für jedes zusätzlich gestartete Programm ein weiterer Druckauftrag gestartet, wodurch beispielsweise unerwünschte Seitenvorschübe oder plötzliche Änderungen im Druckbild auftreten können.

PRINT_TIMEOUT

Die hier in Sekunden eingestellte Zeit wird nach Ende der Druckausgabe abgewartet, bevor ein Druckauftrag als abgeschlossen betrachtet wird. Erfolgt während dieses Zeitraums eine weitere Ausgabe auf den Drucker, werden die Druckdaten dem selben Druckauftrag angefügt und die Zeitzählung beginnt von neuem. Der aktuelle Druckauftrag wird in jedem Fall abgeschlossen, wenn die DOS-Sitzung beendet oder die Tastenkombination [Strg]-[Alt]-[Druck] betätigt wird. Der Wert für *PRINT_TIMEOUT* kann zwischen 0 Sekunden und 3600 Sekunden (1 Stunde) in Schritten von einer Sekunde eingestellt werden. Die Voreinstellung liegt bei 15 Sekunden. Werden 0 Sekunden eingestellt, so wird ein Druckauftrag nur durch das Ende der Sitzung oder Betätigen von [Strg]-[Alt]-[Druck] abgeschlossen.

8.3.5 DOS-Bildschirmeinstellungen

Am Kürzel *VIDEO* sind die DOS-Bildschirmeinstellungen zu erkennen.

VIDEO_8514A_XGA_IOTRAP

Diese Einstellung kontrolliert, auf welche Weise eine DOS-Anwendung auf die Grafikkarte zugreifen darf. Um grafischen DOS-Programmen direkten und damit schnelleren Zugriff auf die Grafikkarte zu gewähren, sollte hier *Off* ausgewählt werden. Dies ist unter anderem bei Spielen sinnvoll. Wird *On* eingestellt, werden Zugriffe auf die Grafikkarte von OS/2 überwacht. Die Voreinstellung ist *On*.

Die Einstellung hat auf DOS-Sitzungen, die im Fenster oder im Hintergrund ablaufen, keinen Einfluß.

VIDEO_FASTPASTE

Ist hier *On* eingeschaltet, beschleunigt OS/2 die Übergabe von Daten aus der Zwischenablage an die DOS-Anwendung. Bei Anwendungen, die mit der schnellen Datenübertragung nicht zurecht kommen, sollte die Voreinstellung *Off* gewählt werden.

Diese Einstellung hat keine Auswirkungen auf DOS-Gesamtbildschirmsitzungen.

VIDEO_MODE_RESTRICTION

Eine Spezialität von OS/2 ist die Erweiterung des konventionellen DOS-Speichers durch ungenutzte Speicherbereiche der Grafikkarte. Falls eine DOS-Anwendung nur geringe Grafikauflösungen oder überhaupt keine Grafik verwendet, kann der dadurch unbenutzte Speicher dem DOS-Speicher zugeschlagen werden. Setzt man *VIDEO_MODE_RESTRICTION* auf *MONO*, wird der DOS-Speicher um 64 Kbyte vergrößert. Wählt man die Einstellung *CGA* aus, sind es sogar 96 Kbyte. Die Voreinstellung lautet *None*, also »keine Einschränkung«.

```
DOS Fenster
 DOS        Strg+Esc = Fensterliste          HELP = Hilfe
C:\>mem


    753664 Byte Gesamtspeicher
    753664 Byte verfügbar für DOS
    740192 größtmögliche Programmgröße

  33947648 Byte EMS-Speicher
  33406976 Byte freier EMS-Speicher

  16711680 Byte XMS-Speicher
  16711680 Byte verfügbarer XMS-Speicher
         0 Byte zusammenhängender EMS-Speicher verfügbar
           DOS resident im HMA (High Memory Area)
           HMA (High Memory Area) in Benutzung


C:\>
```

Abb. 8.3: Ein DOS-Fenster mit eingeschalteter Videomodus-Beschränkung (CGA)

Etwas Vorsicht sollte man bei dieser Einstellung allerdings walten lassen: Die DOS-Anwendung wird nicht daran gehindert, trotz Einschränkung der verfügbaren Grafikstandards in höhere Grafikmodi wie beispielsweise VGA oder SVGA umzuschalten. Dabei können Daten im zusätzlichen DOS-Speicher überschrieben werden, im schlimmsten Fall stürzt die laufende Anwendung ab. Die Einstellung sollte daher nur dann verändert werden, wenn absolut sicher ist, daß die betreffende Sitzung niemals einen hochauflösenden Grafikmodus benutzen wird.

VIDEO_ONDEMAND_MEMORY

Schaltet man aus einer DOS-Gesamtbildschirmsitzung zu einer anderen Anwendung um, muß der Inhalt des Bildschirms gespeichert werden. Der hierfür notwendige Speicherbereich kann entweder beim Start der Sitzung oder beim Umschalten bereitgestellt werden. Welcher Zeitpunkt gewählt wird, kann bei *VIDEO_ONDEMAND_MEMORY* eingestellt werden. Falls Sie die Anforderung dieses Puffers auf den Zeitpunkt des Umschaltens festlegen möchten, stellen Sie *VIDEO_ONDEMAND_MEMORY* auf den Wert *On*. Dies entspricht auch der Voreinstellung. Bei Rechnern mit wenig Arbeitsspeicher kann dadurch ein wenig Speicherplatz freigehalten werden. Ist ausreichend Arbeitsspeicher vorhanden, kann die Einstellung auf *Off* geändert werden, wodurch der Umschaltvorgang etwas schneller vonstatten geht.

Diese Einstellung hat bei DOS-Fenstersitzungen keine Auswirkung.

VIDEO_RETRACE_EMULATION

Von DOS-Anwendungen wird häufig das Retrace-Signal der Grafikkarte als Zeitgeber verwendet. Dadurch können Bildschirmausgaben praktisch flimmerfrei erzeugt werden, was besonders bei Programmen von Bedeutung ist, die grafische Animationen darstellen. Bei solchen Anwendungen, vorzugsweise bei Spielen, sollte *VIDEO_RETRACE_EMULATION* auf *Off* eingestellt sein, damit das echte Retrace-Signal an die Anwendung weitergeleitet wird. Dies ist auch die Voreinstellung. Bei machen Programmen kann die Änderung dieser Einstellung auf *On* zu einer Beschleunigung der Ausgabe auf dem Bildschirm führen. In diesem Fall wird das Retrace-Signal von OS/2 mit einer höheren Frequenz simuliert. Diese Emulation des Retrace-Signals kostet allerdings ein wenig Rechenzeit.

Diese Einstellung hat keine Auswirkung auf DOS-Sitzungen, die im Hintergrund ablaufen.

VIDEO_ROM_EMULATION

Zur Unterstützung bestimmter Funktionen besitzen moderne Grafikkarten ROM-Bausteine, in denen Programmcode gespeichert ist, der von Anwendungen aufgerufen werden kann. Da ROM-Module relativ langsam sind, kann OS/2 solche Funktionen selbst bereitstellen, wodurch diese Aufrufe schneller bearbeitet werden können. Wird die Übernahme dieser Funktionalität in der betreffenden DOS-Sitzung gewünscht, muß hier *On* eingestellt werden. Sind die Funktionen der Grafikkarte selbst ausreichen schnell oder gar schneller, kann man hier *Off* einstellen. Die Voreinstellung ist *On*.

Diese Einstellung hat keine Auswirkung auf DOS-Fenstersitzungen.

VIDEO_SWITCH_NOTIFICATION

Stellen Sie hier *On* ein, wenn die in der betreffenden Sitzung ablaufende Anwendung über das Umschalten von Fenster- zu Gesamtbildschirm-Darstellung informiert werden soll. Einige wenige Anwendungen benutzen dieses Signal, um den Bildschirm beim Umschalten selbst zu speichern und wiederherzustellen. Hier sollte die Voreinstellung *Off* belassen werden, wenn nicht die Dokumentation der betreffenden Anwendung eine Änderung empfiehlt.

VIDEO_WINDOW_REFRESH

Geben Sie hier an, in welchen Abständen die Darstellung eines DOS-Fensters aktualisiert werden soll. Die Zeit zwischen zwei Aktualisierungen kann in Schritten von 1 Zehntelsekunde im Bereich von 0 bis 600 Zehntelsekunden (1 Minute) eingestellt werden. Die Vorgabe von 1 Zehntelsekunde muß in der Regel nicht verändert werden.

Diese Einstellung hat keinen Einfluß auf Aktualisierungen, die durch Tastatureingaben oder durch Rollen des Bildes ausgelöst werden. Bei DOS-Gesamtbildschirmsitzungen ist die Einstellung ebenfalls ohne Bedeutung.

8.3.6 Andere DOS-Einstellungen

Diejenigen Einstellungen, die in keine andere Gruppe passen, finden Sie hier:

AUDIO_ADAPTER_SHARING

Diese Einstellung ist nur dann vorhanden, wenn der Multimedia Presentation Manager »MMPM/2« installiert ist. Sie dient zur Steuerung des Zugriffs auf eine Audio-Karte.

Die Voreinstellung *Required* legt fest, daß die betreffende Anwendung unbedingt Zugriff auf die Audio-Karte benötigt. Ist dieser Wert eingestellt, erscheint eine Fehlermeldung, wenn ein Programm auf die Audio-Karte zugreifen will, während diese bereits von einer anderen Anwendung verwendet wird.

Wird hier *Optional* eingestellt, kann eine Anwendung in dieser DOS-Sitzung den Zugriff auf die Audio-Karte erhalten, wenn diese gerade unbenutzt ist.

Mit der Einstellung *None* wird festgelegt, daß in dieser DOS-Sitzung keine Audio-Karte benötigt wird.

COM_DIRECT_ACCESS

Der Zugriff auf die seriellen Schnittstellen des Rechners erfolgt in DOS-Sitzungen normalerweise über den Treiber »vcom.sys« und unterliegt dadurch der Kontrolle durch OS/2. Diese Kontrolle führt bei manchen Kommunikationsanwendungen, sehr häufig gerade bei Fax-Programmen, zu Problemen. Falls solche Schwierigkeiten auftreten, sollte man versuchen, *COM_DIRECT_ACCESS* auf *On* zu schalten. Dadurch wird der DOS-Anwendung der direkte Zugriff auf die serielle Schnittstelle gewährt. Die Voreinstellung ist *Off*, das heißt, der Zugriff wird durch OS/2 kontrolliert.

COM_HOLD

Als Multitasking-Betriebssystem muß OS/2 auch bei den seriellen Schnittstellen auf konkurrierende Zugriffe achten. Beendet ein Programm den Zugriff auf eine dieser Schnittstellen, wird der betreffende Anschluß wieder in einen definierten Zustand versetzt, damit eine andere Anwendung darauf zugreifen kann. Manche DOS-Kommunikationsanwendungen starten allerdings verschiedene Programme für verschiedene Phasen der Kommunikation, beispielsweise für die Anwahl, den Login-Vorgang und die Datenübertragung. Unter OS/2 hätte dies zur Folge, daß die Verbindung bei jedem neuen Programmstart zusammenbrechen würde, da das Betriebssystem kurzzeitig die Kontrolle über die Schnittstelle übernimmt und sie sofort zurücksetzt. Tritt dieses Verhalten bei einer DOS-Anwendung auf, muß *COM_HOLD* auf *On* gesetzt werden. Dies hat zur Folge, daß die DOS-Sitzung den Zugriff auf eine Schnittstelle vom ersten Zugriff bis zum Ende der Sitzung behält. Die Voreinstellung für *COM_HOLD* ist *Off*.

COM_RECEIVE_BUFFER_FLUSH

Hier wird eingestellt, was mit dem Inhalt des Empfangspuffers der seriellen Schnittstellen passieren soll, wenn die DOS-Sitzung in den Vordergrund geschaltet wird. Die Voreinstellung *None* bewirkt keine besondere Behandlung. Ist *Receive Data Interrupt Enable* eingestellt,

wird der Puffer nicht geleert, wenn die Anwendung die Empfangs-Interrupts aktiviert hat. Ist *Switch to Foreground* ausgewählt, wird der Empfangspuffer beim Umschalten geleert. Die Auswahl *All* entspricht der Kombination der beiden vorigen Einstellungsmöglichkeiten.

COM_SELECT

Die meisten DOS-Anwendungen belegen nur diejenigen Ressourcen, die auch tatsächlich zum Programmablauf benötigt werden. Manche DOS-Kommunikationsprogramme beanspruchen aber bereits beim Programmstart Zugriff auf alle vorhandenen seriellen Schnittstellen. Ist eine solche Anwendung aktiv, kann kein anderes Programm auf die COM-Schnittstellen zugreifen. Um dem abzuhelfen, kann bei der Einstellung *COM_SELECT* ausgewählt werden, welche serielle Schnittstelle von der Anwendung verwendet werden darf. Die Voreinstellung *All* erlaubt den Zugriff auf alle seriellen Schnittstellen, die Einstellung *None* sperrt alle COM-Schnittstellen für die betreffende DOS-Sitzung.

DOS_AUTOEXEC

Tragen Sie hier den Namen einer Stapeldatei ein, die beim Start der DOS-Sitzung automatisch ausgeführt werden soll. Die Voreinstellung lautet »x:\autoexec.bat« (wobei mit »x:« das OS/2-Installationslaufwerk bezeichnet ist). Mit dieser Funktion kann man beispielsweise in einer bestimmten DOS-Sitzung ein spezielles TSR-Programm starten, das nur in dieser DOS-Umgebung benötigt wird.

DOS_BACKGROUND_EXECUTION

Hier können Sie die Parallelverarbeitung der DOS-Sitzung abschalten, wenn sich diese im Hintergrund befindet. Dies ist dann sinnvoll, wenn der Programmablauf vom Benutzer überwacht werden muß (vorzugsweise bei Spielen) oder wenn ein Programm zumeist unproduktiv ist und dadurch im Hintergrund unnütz Rechenzeit beanspruchen würde. Die Voreinstellung *On* gestattet den Programmlauf im Hintergrund. Um die Verarbeitung im Hintergrund zu verhindern, stellen Sie *Off* ein.

DOS_BREAK

Diese Einstellung entspricht dem Eintrag *BREAK=ON* bzw. *BREAK=OFF* in der »config.sys« von DOS. Ist hier *On* eingestellt, wird bei jedem DOS-Aufruf überprüft, ob eine der Tastenkombinationen [Strg]-[C] oder [Strg]-[Untbr] betätigt wurde. Die Voreinstellung *Off* bewirkt, daß ein Programmabbruch lediglich bei Dateioperationen möglich ist. Bei Änderung dieser Einstellung sollte bedacht werden, daß die Abfrage der beiden Tastenkombinationen etwas Rechenzeit kostet. Man sollte hier also nur dann *On* einstellen, wenn die Möglichkeit des Programmabbruchs bei der betreffenden Anwendung unbedingt notwendig ist und in der Anwendung selbst nicht oder nicht ausreichend vorgesehen ist.

DOS_DEVICE

Gerätetreiber werden unter DOS üblicherweise in der Config.sys-Datei mit der Anweisung *DEVICE=...* geladen. Bei einer DOS-Sitzung unter OS/2 ist die Einstellung *DOS_DEVICE* für das Laden solcher Treiber zuständig. Geben Sie einfach im Eingabefeld den Dateinamen mit Verzeichnis ein. Falls Sie mehrere Treiber laden möchten, verwenden Sie für jeden Datei-

namen eine eigene Zeile im Eingabefeld. Die hier eingetragenen Treiber werden nur in der betreffenden Sitzung geladen. Soll ein Treiber in allen DOS-Sitzungen aktiv sein, tragen Sie ihn – wie unter DOS – mit der Anweisung *DEVICE=...* in der »config.sys« von OS/2 ein.

DOS_FCBS

Ältere DOS-Anwendungen verwenden zur Dateiverwaltung die sogenannten File Control Blocks (FCBs). Die Anzahl der in der DOS-Sitzung verwendbaren FCBs kann unter *DOS_FCBS* eingestellt werden. Die Anzahl der FCBs kann zwischen 0 und 255 FCBs in Schritten einzelner FCBs eingestellt werden. Die Voreinstellung liegt bei 16 FCBs.

Diese Einstellung muß normalerweise niemals verändert werden, da FCBs überwiegend von Anwendungen benutzt werden, die für DOS-Versionen vor DOS 2.0 geschrieben wurden. Jüngere Anwendungen verwenden eine andere Art von Datenstrukturen zur Dateiverwaltung.

DOS_FCBS_KEEP

Sind alle verfügbaren File Control Blocks (FCBs) ausgeschöpft, kann das Betriebssystem beim Öffnen weiterer Dateien bereits benutzte FCBs schließen. Bei *DOS_FCBS_KEEP* wird angegeben, wieviele der unter *DOS_FCBS* angegebenen FCBs nicht auf diese Weise geschlossen werden können. Der Wert kann auch hier von 0 bis 255 in Einerschritten beeinflußt werden. Die Voreinstellung ist 8 FCBs.

Diese Einstellung braucht normalerweise nicht verändert werden, da FCBs überwiegend von Anwendungen benutzt werden, die für DOS-Versionen vor DOS 2.0 geschrieben wurden. Spätere Anwendungen verwenden eine andere Art von Datenstrukturen zur Dateiverwaltung.

DOS_FILES

Die Einstellung *DOS_FILES* entspricht der Anweisung *FILES=...* in der »config.sys« von DOS und bestimmt, wieviele Dateien in der betreffenden DOS-Sitzung gleichzeitig geöffnet sein können. Die Einstellung kann zwischen 20 und 255 in Einerschritten variiert werden. In der »config.sys« kann, wie unter DOS, mit der Anweisung *FILES=...* ein globaler Vorgabewert festgesetzt werden.

Beachten Sie, daß die Vergrößerung des Wertes für *DOS_FILES* in der jeweiligen DOS-Sitzung Speicherplatz kostet. Die meisten DOS-Anwendungen, die hier einen höheren Wert als 20 benötigen, weisen in ihrer Dokumentation deutlich darauf hin. Fehlen solche Hinweise, sollte der Vorgabewert benutzt werden.

DOS_LASTDRIVE

Wie unter DOS mit der Config.sys-Einstellung *LASTDRIVE* können Sie hier die höchste verfügbare Laufwerksbezeichnung innerhalb der DOS-Sitzung bestimmen. Dies ist beispielsweise dann sinnvoll, wenn die ablaufende Anwendung einige Laufwerksbuchstaben für eigene Zwecke reservieren möchte. Im Eingabefeld von *DOS_LASTDRIVE* muß der höchste als Laufwerksbezeichnung zugelassene Buchstabe eingetragen werden. Voreingestellt ist der Buchstabe »Z«, das bedeutet, daß alle Laufwerksbezeichnungen innerhalb der DOS-Sitzung verfügbar sind.

DOS_SHELL

Falls in der betreffenden Sitzung ein alternativer Kommandoprozessor verwendet werden soll, muß hier der Dateiname mit Verzeichnisangabe eingetragen werden. Der Vorgabewert ist »x:\os2\mdos\command.com« (wobei »x:« das OS/2-Installationslaufwerk bezeichnet). Ein Beispiel für einen alternativen Kommandoprozessor ist das Programmpaket 4DOS. Der darin enthaltene Kommandoprozessor ist wesentlich leistungsfähiger und besser konfigurierbar als »command.com«. 4DOS ist als Shareware-Programm erhältlich.

DOS_STARTUP_DRIVE

OS/2 bietet die Möglichkeit, in einer DOS-Sitzung einen anderen als den bei OS/2 mitgelieferten DOS-Kern zu verwenden. Die Einstellung *DOS_STARTUP_DRIVE* bezeichnet den Ort, an dem der gewünschte DOS-Kern zu finden ist. Dies kann zum Beispiel ein Diskettenlaufwerk (»A:«) oder eine primäre Partition auf der Festplatte (»C:«) sein. Eine weitere Möglichkeit ist die Angabe einer mit dem Programm VMDISK<$i erzeugten <$iAbbilddatei einer DOS-Startdiskette. Durch die Verwendung eines anderen DOS-Kerns lassen sich häufig Problem lösen, die durch Beeinflussung der anderen DOS-Einstellungen nicht lösbar waren. Allerdings ist dieser Weg nur bei sehr wenigen Programmen notwendig. Im folgenden Abschnitt dieses Kapitels wird diese Möglichkeit der DOS-Emulation unter OS/2 nochmals ausführlicher behandelt.

DOS_VERSION

Einige DOS-Programme fragen beim Betriebssystem die Versionsnummer ab, um die unterstützten Funktionen bestimmen zu können. Das bei OS/2 mitgelieferte DOS ist zwar weithin kompatibel mit DOS 5.0, liefert allerdings eine etwas ungewöhnliche Versionsmeldung an DOS-Programme zurück. Dadurch verweigern manche Programme den Dienst, obwohl sie problemlos in der DOS-Umgebung von OS/2 ablauffähig wären. Solchen Programmen kann OS/2 vorspielen, es sei eine bestimmte DOS-Version installiert. Dabei wird allerdings nicht die gewünschte Kompatibilität hergestellt, sondern es wird lediglich die Versionsabfrage der betreffenden Anwendung mit der gewünschten Versionsnummer beantwortet. Um diese Funktion für ein bestimmtes Programm einzurichten, muß der bereits vorhandenen Liste bei *DOS_VERSION* eine Zeile in folgendem Format hinzugefügt werden:

```
Name der Exe-Datei,Vorkommastelle,Nachkommastelle,255
```

Um beispielsweise dem Programm dBase IV die DOS-Version 5.0 vorzugaukeln, müßte man folgenden Eintrag vornehmen:

```
DBASE.EXE,5,0,255
```

HW_NOSOUND

Mit dieser Einstellung läßt sich die Ausgabe von Tönen innerhalb der betreffenden DOS-Sitzung abschalten. Die Voreinstellung *Off* gestattet die Ausgabe von Tönen, während sie durch die Einstellung *On* unterbunden wird. Die Ausgabe des Bell-Zeichens mit dem ASCII-Wert 7 wird durch *HW_NOSOUND* nicht beeinflußt.

HW_ROM_TO_RAM

Wird *HW_ROM_TO_RAM* auf *On* geschaltet, wird der Inhalt des ROM-BIOS für diese DOS-Sitzung in den gewöhnlichen Arbeitsspeicher kopiert. Dadurch wird die Verarbeitung der BIOS-Funktionen beschleunigt. Außerdem ist es dann möglich, mit einem Debugger auch im BIOS-Bereich zu arbeiten. Allerdings können die BIOS-Inhalte dann verändert werden, wodurch die DOS-Sitzung abstürzen könnte. Aus Sicherhetsgründen lautet daher die Voreinstellung *Off*.

HW_TIMER

Jeder PC verfügt über eine Reihe von Zeitgebern, mit denen man die Geschwindigkeit von Abläufen unabhängig vom Prozessor und dessen Taktfrequenz festlegen kann. Mit der Einstellung *HW_TIMER* kann ausgewählt werden, ob diese Zeitgeber von OS/2 kontrolliert werden sollen, oder ob der ablaufenden DOS-Anwendung der Zugriff auf die Hardware-Zeitgeber gestattet werden soll. Die Voreinstellung *Off* bewirkt die Kontrolle der Zeitgeber durch OS/2. Bei zeitkritischen Programmen, insbesondere bei Spielen oder Kommunikationsanwendungen, ist es häufig notwendig, *On* einzustellen.

IDLE_SECONDS

Im präemptiven Multitasking von OS/2 laufen alle Anwendungen zunächst mit derselben Priorität, das heißt, alle Anwendungen haben das gleiche Maß an Rechenzeit zur Verfügung. Bei DOS-Programmen kommt es allerdings häufig vor, daß Rechenzeit unnötig für Tastaturabfragen verschwendet wird. OS/2 versucht, solche Situationen zu erkennen und verringert dann die Priorität der betreffenden Anwendung. Unter gewissen Umständen schlägt diese Erkennung allerdings fehl und eine Anwendung bekommt fälschlicherweise zu wenig Rechenzeit, um flüssig ablaufen zu können. Die Einstellung *IDLE_SECONDS* kann in solchen Fällen Abhilfe schaffen. Der Wert gibt an, wieviele Sekunden einer Anwendung das vermeintliche Verschwenden von Rechenzeit gestattet werden soll, bevor die Priorität verringert wird. Die Voreinstellung liegt beim Minimalwert von 0 Sekunden, das bedeutet, daß OS/2 die Priorität der DOS-Sitzung sofort verringert, wenn zu oft die Tastatur abgefragt wird. Der Maximalwert für *IDLE_SECONDS* liegt bei 60 Sekunden.

Häufig wird diese Einstellung wichtig, wenn eine Anwendung auf Daten von einem sehr langsamen Gerät warten muß. Falls OS/2 in einer solchen Situation die Priorität für die Anwendung verringert, kann es zu Fehlern bei der anschließenden Datenübertragung kommen. Eine weitere Anwendung ist ein Phänomen, das vereinzelt in der Win-OS/2-Umgebung auftritt: Der Mauszeiger bewegt sich nur ruckartig. Versuchen Sie in diesem Fall, für *IDLE_SECONDS* einen höheren Wert, beispielsweise 5 Sekunden, einzustellen.

IDLE_SENSITIVITY

Diese Einstellung setzt einen Schwellenwert für die Leerlauffeststellung fest. Wird die Tastatur mit einer höheren Frequenz abgefragt, als dies bei *IDLE_SENSITIVITY* eingestellt wurde, wird die Anwendung als im Leerlauf befindlich betrachtet. Der Bereich von 0 bis 100 ist dabei als Bruchteil einer in OS/2 festgelegten Maximalfrequenz anzusehen. Setzt man den Maximalwert 100 fest, wird die Leerlauffeststellung für diese Sitzung vollständig abgeschaltet. Die Voreinstellung liegt bei 75.

Sind in einem OS/2-System mehrere DOS-Anwendungen aktiv, sollte man ein wenig mit *IDLE_SENSITIVITY* experimentieren, um eine optimale Zuweisung der Rechenzeit zu erreichen. Falls eine Anwendung nicht flüssig abläuft, sollte der Wert für *IDLE_SENSITIVITY* erhöht werden, bis das Programm mit vernünftiger Geschwindigkeit abläuft. Damit für andere Anwendungen mehr Rechenzeit zur Verfügung steht, sollte *IDLE_SENSITIVITY* bei weniger wichtigen DOS-Sitzungen so niedrig wie möglich angesetzt werden.

INT_DURING_IO

Während Datei-Operationen (Ein- oder Ausgabe) können normalerweise keine Unterbrechungsanforderungen bearbeitet werden. Diese werden aber häufig benutzt, um den flüssigen Ablauf zeitkritischer Vorgänge zu gewährleisten. Das dabei entstehende Problem kann man vor allem bei DOS-Multimedia-Anwendungen unter OS/2 beobachten: ruckartige Bildschirmanzeige, stotternde Audio-Ausgabe. Um diesem Problem abzuhelfen, sollte man *INT_DURING_IO* auf *On* setzen. Dadurch wird innerhalb der DOS-Sitzung ein zweiter Thread für Dateioperationen erzeugt, so daß Ein-/Ausgabe- und Unterbrechungsanforderungen gleichzeitig bearbeitet werden können. Da durch den zusätzlichen Thread mehr Speicher verbraucht wird, sollte man bei gewöhnlichen Anwendungen die Voreinstellung *Off* verwenden.

SESSION_PRIORITY

Mit dieser Einstellung können Sie die Vorgabe für die Priorität der DOS-Sitzung bestimmen. Der Wert kann von 1 (niedrigste Priorität) bis 32 (höchste Priorität) festgelegt werden. Die Voreinstellung ist 1.

8.3.6 Win-OS/2-Einstellungen

Neben allen DOS-Einstellungen verfügen Windows-Programme zusätzlich über die folgenden Einstellungsmöglichkeiten:

WIN_RUN_MODE

Hier können Sie bestimmen, in welchem Betriebsmodus die Windows-Umgebung gestartet wird. Zur Auswahl stehen *3.1 Standard* und *3.1 Enhanced Compatibility*. Die Voreinstellung ist *3.1 Enhanced Compatibility*, da bei dieser Einstellung manche Programme schneller ablaufen. Viele Anwendungen sind im anderen Modus überhaupt nicht ablauffähig.

WIN_DDE

Diese Einstellung aktiviert oder deaktiviert den dynamischen Datenaustausch (Dynamic Data Exchange, DDE) zwischen der Windows-Sitzung und OS/2-Anwendungen. Innerhalb einer Windows-Umgebung kann der dynamische Datenaustausch in jedem Fall verwendet werden. Die Voreinstellung ist *Off*; es kann damit kein dynamischer Datenaustausch zwischen der Windows-Sitzung und OS/2-Anwendungen erfolgen. Um Speicherplatz zu sparen, sollte diese Einstellung nur dann geändert werden, wenn DDE mit der betreffenden Anwendung tatsächlich verwendet werden soll.

WIN_CLIPBOARD

Sowohl das OS/2-System als auch das Windows-System verfügen über eine Zwischenablage zum Austausch von Daten zwischen zwei oder mehreren Anwendungen. OS/2 kann für jede Windows-Umgebung eine eigene Zwischenablage erzeugen, die für andere Anwendungen nicht zugänglich ist. In diesem Fall ist allerdings kein Austausch von Daten zwischen der betreffenden Windows-Sitzung und DOS- , OS/2- oder anderen Windows-Sitzungen möglich. Um Arbeitsspeicher zu sparen und die Startzeit der Windows-Umgebung zu verkürzen, sollte hier *Off* eingestellt werden, damit die Windows-Umgebung ihre eigene Zwischenablage verwendet. Dies ist auch die Voreinstellung. Um den Datenaustausch mit anderen Sitzungen zu ermöglichen, stellen Sie *On* ein.

WIN_ATM

Setzt man hier *On*, können in der Windows-Umgebung neben TrueType-Schriften auch Postscript-Schriften vom Typ 1 verwendet werden. Zur Unterstützung dieser Funktion enthält OS/2 eine Windows-Version des Adobe Type Managers, mit dem die Postscript-Schriften verwaltet werden. Die Voreinstellung ist *Off*, da durch das Laden des Adobe Type Managers Speicherplatz belegt und die Startzeit der Windows-Umgebung erhöht wird. Aus diesem Grund sollte man diese Einstellung nur dann verändern, wenn man in der betreffenden Sitzung die qualitativ hochwertigeren Postscript-Schriftarten nutzen möchte.

8.4 Die letzte Möglichkeit: DOS Virtual Machine Boot

Der überwiegende Großteil der DOS-Anwendungen läuft auf Anhieb problemlos in der DOS-Umgebung von OS/2. Der größte Teil der übrigen Programme ist nach Änderung der DOS-Einstellungen lauffähig. Einige DOS-Programme sind unter OS/2 nicht ablauffähig, weil sie die Systemstabilität beeinträchtigen würden (z.B. Programme, die Speichermanager nach dem VCPI-Standard verwenden). Es gibt noch eine Klasse von Programmen, die in keine dieser Kategorien fällt. Diese Programme wären prinzipiell unter OS/2 ablauffähig, benutzen aber spezielle Einrichtungen des DOS-Betriebssystems, die in der DOS-Umgebung von OS/2 nicht zur Verfügung stehen. Als Beispiele seien die Peer-Netzwerke »Netware Lite« und »Personal Netware« von Novell genannt. Solche Programme, auch die beiden genannten, können mit einer speziellen Einrichtung von OS/2 zum Laufen gebracht werden: Mit dem sogenannten »Virtual Machine Boot«.

8.4.1 DOS von Laufwerk A:

Die Idee ist einfach: Wenn ein Programm mit dem DOS-Kern von OS/2 nicht zurechtkommt, dann bietet man ihm eben einen anderen DOS-Kern an. Vorzugsweise ist hier ein DOS-Kern zu verwenden, dessen Kompatibilität zu diesem fraglichen Programm bekannt ist. Eine Möglichkeit, eine solche DOS-Version unter OS/2 zu verwenden, finden Sie im Befehlszeilen-Ordner: *DOS von Laufwerk A:*. Liegt eine DOS-Startdiskette im Diskettenlaufwerk »A:«, braucht man nur dieses Objekt zu öffnen, um die DOS-Version, die sich auf der Startdiskette befindet, in einem Fenster zu starten.

Diese Funktionalität wird mit der DOS-Einstellung *DOS_STARTUP_DRIVE* realisiert. Beim Objekt *DOS von Laufwerk A:* werden Sie bei dieser Einstellung den Eintrag *A:* vorfinden. Bevor wir diese DOS-Einstellung näher beleuchten, wenden wir uns kurz den Besonderheiten einer solchen Bootdiskette zu:

– Soll in der DOS-Sitzung XMS-Speicher verwendet werden, ist der Treiber »\os2\mdos\himem.sys« statt des üblichen DOS-Treibers zu verwenden.

– Soll in der DOS-Sitzung EMS-Speicher verwendet werden, so ist der Treiber »\os2\mdos\emm386.sys« statt des üblichen DOS-Treibers zu verwenden.

– Wird Maus-Unterstützung benötigt, so muß in der »autoexec.bat« der Maustreiber »\os2\mdos\mouse.com« aufgerufen werden.

– Falls innerhalb der DOS-Sitzung auf HPFS-Partitionen zugegriffen werden soll, muß der Treiber fsfilter.sys auf die Startdiskette kopiert werden. In der »config.sys« der Startdiskette ist dann in der ersten Zeile der Eintrag

  ```
  DEVICE=FSFILTER.SYS
  ```

  zu ergänzen.

Wurden die Konfigurationsdateien der Startdiskette entsprechend dieser Hinweise abgeändert, kann diese Diskette mit dem Objekt *DOS von Laufwerk A:* verwendet werden. Falls OS/2 auf einem logischen Laufwerk installiert ist und der Zugriff auf eine primäre Partition mit installiertem DOS-Betriebssystem möglich ist, kann diese Partition auf dieselbe Weise wie eine Startdiskette verwendet werden. Bei der Einstellung *DOS_STARTUP_DRIVE* ist dann der Laufwerksbuchstabe der DOS-Partition einzutragen.

8.4.2 Virtual Machine Boot mit Abbilddatei

Leider dauert es recht lange, bis ein DOS-Betriebssystem von Diskette gestartet ist. Um den Startvorgang zu beschleunigen, gibt es unter OS/2 die Möglichkeit, von einer DOS-Startdiskette eine Abbilddatei zu erzeugen, von der das DOS-System dann wesentlich schneller gestartet werden kann.

Hat man, wie im vorigen Abschnitt beschrieben, eine DOS-Startdiskette erzeugt, die mit dem Objekt *DOS von Laufwerk A:* funktioniert, kann man mit dem Programm VMDisk aus der Diskette eine Abbilddatei erzeugen:

```
VMDISK A: ABBILD.IMG
```

Dabei steht »A:« für das Diskettenlaufwerk, in dem sich die Startdiskette befindet. Der Name der Abbilddatei, hier »abbild.img«, kann frei gewählt werden. Den Namen der Abbilddatei und das Verzeichnis, in dem sie abgelegt ist, kann man nun in das Eingabefeld bei *DOS_STARTUP_DRIVE* eintragen, und schon wird in der betreffenden DOS-Sitzung der auf der Startdiskette enthaltene DOS-Kern geladen.

Leider hat der größere Komfort der Abbilddatei auch einen Pferdefuß: In der DOS-Sitzung, die mit der Abbild-Datei gestartet wurde, ist das Diskettenlaufwerk »A:« nicht mehr ansprechbar. Es gibt zwar ein Laufwerk mit dem Namen »A:«, es handelt sich dabei allerdings um ein virtuelles Laufwerk, das die Abbilddatei repräsentiert. Auf diesem virtuellen Laufwerk – also auf der Abbilddatei – kann man alle Operationen durchführen, die auch auf der Startdiskette zur Verfügung stehen. Es ist damit also möglich, die Konfigurationsdateien »config.sys« und »autoexec.bat« dieser Sitzung zu laden, abzuändern und wieder zu speichern. Es ist aber auch möglich, Dateien zu löschen oder sogar das Laufwerk zu formatieren, wodurch die Abbilddatei unbrauchbar wird.

Es gibt aber auch hierfür eine Lösung. Das Programm FSAccess, es liegt im Verzeichnis »\os2\mdos« auf dem OS/2-Installationslaufwerk, kann den Zugriff auf das echte Laufwerk »A:« ermöglichen. Durch die Anweisung

```
FSACCESS A
```

wird der Zugriff auf das echte Diskettenlaufwerk A: freigegeben. Der Zugriff auf die Abbilddatei ist ab diesem Zeitpunkt nicht mehr möglich. Mit FSAccess ist es auch möglich, das echte Laufwerk A: auf einen freien Laufwerksbuchstaben umzuleiten. Der Befehl hierzu lautet

```
FSACCESS X=A
```

wobei X für einen freien Laufwerksbuchstaben steht, unter dem ab der Ausführung des Befehls auf das echte Laufwerk »A:« zugegriffen werden kann.

8.5 Besonderheiten der DOS-Umgebung

8.5.1 Die Befehlszeilen-Objekte

Statt ein Programm über ein zugehöriges Programm-Objekt zu starten, kann man auch den von DOS bekannten Weg über die Befehlszeile gehen. Im Ordner *OS/2 System* befindet sich der Ordner *Befehlszeilen*, der eine Reihe von Befehlszeilen-Objekten enthält. Diese Programmobjekte sind mit Ausnahme des Objekts *Win-OS/2 Fenster* auf dieselbe Weise realisiert: Im Einstellungsnotizbuch ist im Eingabefeld *Pfad und Dateiname* nur ein Stern eingetragen und auf der Sitzung-Seite ist der gewünschte Sitzungs-Typ ausgewählt. Möchte man weitere Befehlszeilen-Objekte erstellen, beispielsweise mit speziellen Speichereinstellungen, kann man nach genau demselben Muster vorgehen.

Die einzige Ausnahme im Befehlszeilen-Ordner ist das Objekt *Win-OS/2 Fenster*. In diesem Fall wird einfach der Windows Programm-Manager im Seamless-Modus auf der OS/2-Arbeitsoberfläche ausgeführt. Im Einstellungsnotizbuch ist dementsprechend als Pfad und Dateiname die Exe-Datei des Programm-Managers »progman.exe« eingestellt, während auf der Sitzung-Seite die Auswahl Win-OS/2-Fenster aktiviert ist.

8.5.2 Die OS/2-Zwischenablage in DOS-Sitzungen

IBM plaziert OS/2 im Markt als Integrationsplattform, die OS/2-, DOS- und Windows-Programme unter einen Hut bringt. Diese Integration ist auch bei der Zwischenablage gelungen, so daß man sehr komfortabel Daten zwischen Anwendungen austauschen kann. Selbst die funktionell unterlegenen DOS-Programme kommen beim Datenaustausch via Zwischenablage nicht zu kurz.

Die Steuerung der Zwischenablage in DOS-Sitzungen erfolgt über das Systemmenü des DOS-Fensters. Läuft eine DOS-Sitzung im Gesamtbildschirm ab, muß zunächst mit der Tastenkombination [Alt]-[Pos1] in die Fensterdarstellung umgeschaltet werden. Später kann auf dieselbe Weise wieder in den Gesamtbildschirm umgeschaltet werden. Das Systemmenü erhält man, indem man mit der Maus den Knopf in der linken oberen Fensterecke betätigt.

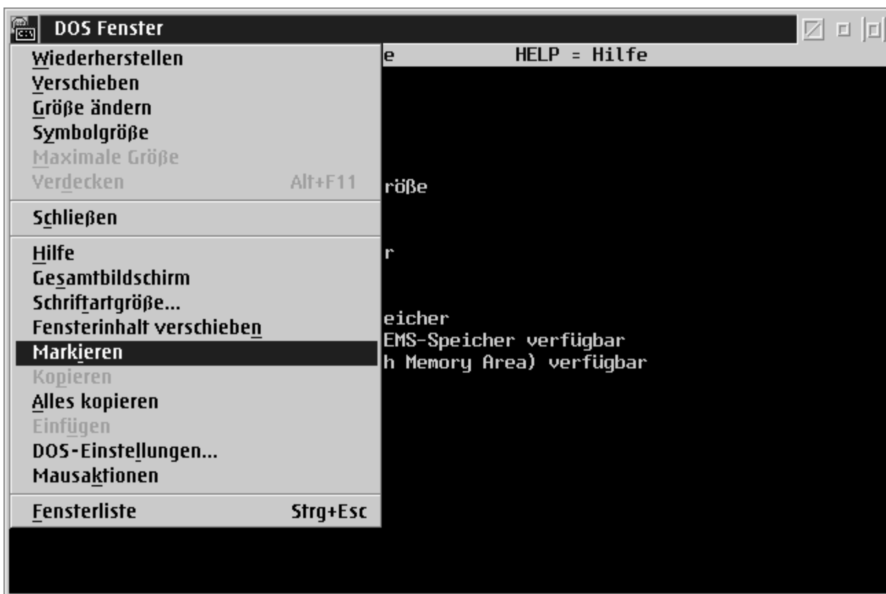


Abb. 8.4: Das Systemmenü einer DOS-Sitzung

Folgende Menüpunkte sind momentan für uns interessant:

– Markieren

– Kopieren bzw. Alles kopieren

– Einfügen

– Mausaktionen

8.5.3 Markieren

Markieren bezeichnet die Auswahl eines rechteckigen Fensterbereichs, der dann in die Zwischenablage kopiert wird. Wird dieser Menüpunkt ausgewählt, verändert sich das Aussehen des Mauszeigers. Bewegen Sie dann den Mauszeiger auf eine Ecke des gewünschten Ausschnitts und halten die linke Maustaste gedrückt. Beim Ziehen der Maus wird nun der ausgewählte Bereich hervorgehoben dargestellt. Wenn Sie mit Ihrer Auswahl zufrieden sind, lassen Sie die Maustaste los. Sie können diesen Vorgang so oft wiederholen, bis Sie das richtige Rechteck getroffen haben. Der Vorgang kann durch Betätigen von [Esc] oder durch nochmalige Auswahl des Menüpunkts *Markieren* abgebrochen werden, der Inhalt der Zwischenablage bleibt dann unverändert.



Abb. 8.5: Markieren in einem DOS-Fenster

8.5.4 Kopieren

Der Menüpunkt *Kopieren* im Systemmenü veranlaßt, daß der markierte Bereich in die Zwischenablage übertragen wird. Dieser Menüpunkt ist nur dann aktiv, wenn zuvor eine Markierung im Fenster angebracht wurde. Alternativ kann zum Kopieren des ausgewählten Rechtecks in die Zwischenablage auch die Eingabetaste betätigt werden.

Mit dem Menüpunkt *Alles kopieren* wird automatisch der gesamte Fensterbereich in die Zwischenablage kopiert. Dazu muß kein Bereich markiert sein, der Menüpunkt ist daher immer anwählbar.

8.5.5 Einfügen

Einfügen ist das Gegenstück zum Kopieren: Der aktuelle Inhalt der Zwischenablage wird in der DOS-Sitzung eingefügt. Dazu simuliert OS/2 die entsprechenden Tastaturaktionen, die Anwendung hat also den Eindruck, der Text werde über die Tastatur eingegeben. Es ist nicht möglich, in einer DOS-Sitzung Grafik aus der Zwischenablage einzufügen. Der Menüpunkt *Einfügen* ist nur dann aktiv, wenn die Zwischenablage Text enthält.

8.5.6 Mausaktionen

Um den Umgang mit der Zwischenablage zu vereinfachen, wurde in Version 4 von OS/2 eine Möglichkeit geschaffen, die wichtigsten Operationen im DOS-Fenster mit der Maus auszuführen. Diese Funktionalität wird über den Menüpunkt Mausaktionen im Systemmenü gesteuert. Im System-Objekt im Ordner *Systemkonfiguration* kann auf der Seite *Benutzerschnittstelle* eingestellt werden, ob die Mausaktionen beim Start einer DOS-Sitzung ein- oder ausgeschaltet sein sollen. Bei jeder laufenden DOS-Sitzung können die Mausaktionen beliebig über das Systemmenü aktiviert oder deaktiviert werden.

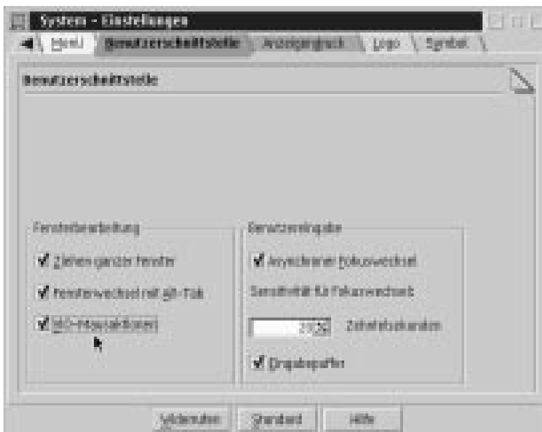


Abb. 8.6: Die Notizbuchseite zum Aktivieren und Deaktivieren der Mausaktionen

Wenn die Mausaktionen in einer DOS-Sitzung aktiviert sind, stehen die wichtigsten Aktionen aus dem Systemmenü auch über ein Kontextmenü zur Verfügung. Dieses Kontextmenü erhal-

ten Sie, wenn Sie mit der rechten Maustaste auf die Fläche des DOS-Fensters klicken. Zusätzlich ist es bei aktivierten Mausaktionen möglich, einen Bereich des Fensters bei gedrückter linker Maustaste zu markieren. Es ist nicht mehr notwendig, im Systemmenü den Punkt *Markieren* auszuwählen.

Neben den Funktionen zur Steuerung der Zwischenablage befinden sich im Kontextmenü des DOS-Fensters Funktionen zum Ändern der verwendeten Schriftartgröße, zum Bearbeiten der DOS-Einstellungen, zum Schließen des Fensters und zum Deaktivieren der Mausaktionen. Letzteres ist beispielsweise dann notwendig, wenn im Fenster eine Anwendung abläuft, die selbst das Ziehen der Maus bei gedrückter linker Maustaste vorsieht.

8.5.7 Das DPMI-1.0-Subset

Neben den Speicherverwaltungsmodellen »EMS« und »XMS« unterstützt die DOS-Umgebung von OS/2 den Standard *DOS Protected Mode Interface* (DPMI). In OS/2 Warp Version 4 ist eine Untermenge der DPMI-1.0-Schnittstelle implementiert, die für die meisten Programme ausreicht. In den Einstellungen einer DOS- oder Windows-Sitzung kann die DPMI-Unterstützung an- und abgeschaltet werden und man kann DPMI-Speicher bis zu einer Größe von 512 Mbyte vergeben. Die Windows-Umgebung unter OS/2 verwendet – anders als unter DOS – solchen DPMI-Speicher. Es ist also wichtig, ausreichend DPMI-Speicher für Windows-Sitzungen einzustellen. Bei anderen Anwendungen kann man der Dokumentation entnehmen, ob DPMI-Speicher benötigt wird.

Einen großen Anteil an den DPMI-fähigen Programmen bilden Anwendungen, die mit den Entwicklungssystemen der Firma Borland erzeugt wurden (unter anderem auch die meisten Borland-Produkte selbst). Mit diesen Anwendungen gibt es unter OS/2 ein kleines Problem: Richtet man für sie wie gewohnt ein Programm-Objekt ein und versucht, dieses zu öffnen, reagiert OS/2 mit der folgenden Meldung (hier am Beispiel Borland Pascal):

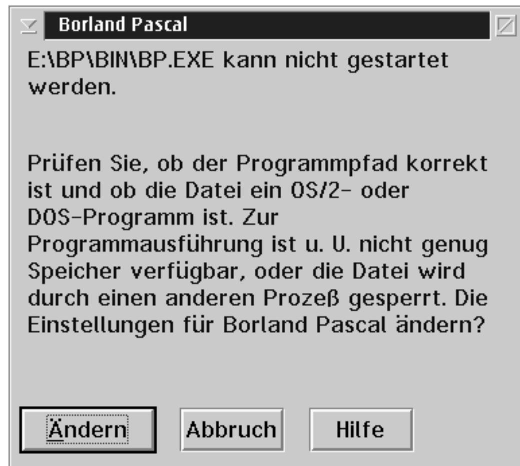


Abb. 8.7: Der Start von Borland Pascal aus einem Programmobjekt

Das Problem ist eine Eigenheit einer bestimmten Klasse von DPMI-Exe-Dateien: Es handelt sich um das Exe-Format von OS/2 1.3 und OS/2 glaubt daher, es handle sich um ein OS/2-Programm. Betrachtet man die Sitzung-Seite im Einstellungsnotizbuch, kann man diese Tatsache an den deaktivierten Einstellungen erkennen. Da für ein OS/2-Programm keine DOS-Umgebung erzeugt wird, kann das Programm nicht gestartet werden. Mit einem kleinen Trick kann man jedoch für solche Programme trotzdem Objekte erzeugen. Dieser Trick beruht auf dem Erzeugen einer Befehlszeile des gewünschten Typs, der dann als Parameter ein Startbefehl für das fragliche Programm mitgegeben wird. Dieser Startbefehl lautet

```
/C Programmname
```

also in unserem Beispiel

```
/C E:\BP\BIN\BP.EXE
```

Die erste Seite im Einstellungsnotizbuch sieht dann folgendermaßen aus:

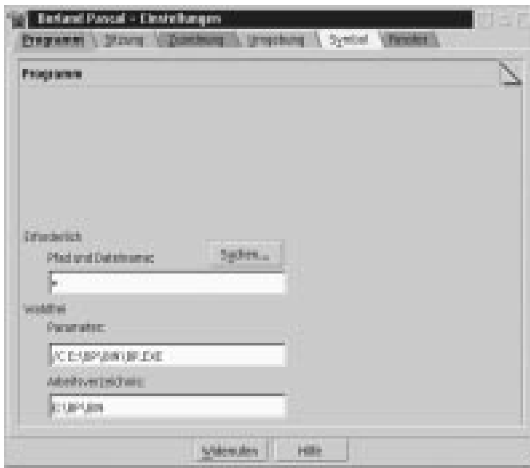


Abb. 8.8: Die passenden Einstellungen für Borland-DPMI-Programme

Nun muß man lediglich noch auf der Sitzung-Seite den Sitzungstyp, also DOS-Gesamtbildschirm oder DOS-Fenster eintragen und auf der Symbol-Seite muß das Symbol manuell eingestellt werden (beispielsweise durch Drag&Drop des gewünschten Symbols auf das bereits eingestellte Symbol).

8.6 Besonderheiten der Win-OS/2-Umgebung

8.6.1 Win-OS/2-Konfiguration

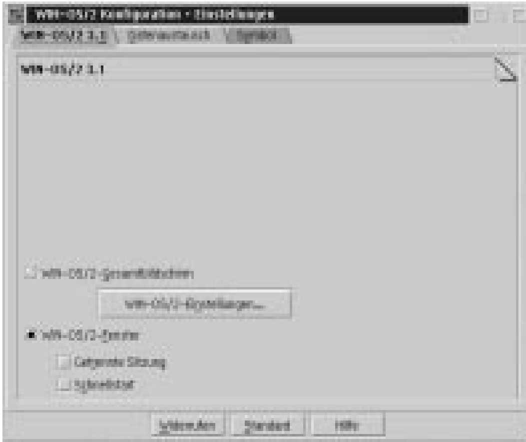


Abb. 8.9: Das Notizbuch für die Win-OS/2-Konfiguration

Im Ordner *Systemkonfiguration* befindet sich das Objekt *Win-OS/2-Konfiguration*. Dieses Objekt erlaubt die Vorgabe von globalen Einstellungen für Windows-Programme. Eine der wichtigsten Einstellungen befindet sich bereits auf der ersten Seite des Notizbuchs: *Win-OS/2 Schnellstart*. Ist diese Option aktiviert, wird schon beim Start des OS/2-Systems unbemerkt eine Windows-Umgebung erzeugt. Wird nun eine Windows-Anwendung gestartet, so kann sie – falls die Win-OS/2-Einstellungen übereinstimmen – in dieser bereits gestarteten Windows-Umgebung ausgeführt werden. Es entfällt dann die Wartezeit für den Start von Win-OS/2.

Die Schnellstart-Option sollte nur verwendet werden, wenn während der Arbeit mit dem Rechner voraussichtlich häufig Windows-Programme gestartet werden. Ansonsten sollte darauf zugunsten eines schnelleren Systemstarts und mehr freien Speichers verzichtet werden.

8.6.2 Der Adobe Type Manager

OS/2 enthält eine sehr nützliche Erweiterung des Windows-Systems, die automatisch installiert wird: den Adobe Type Manager. Mit diesem Programm wird es möglich, neben den TrueType-Schriftarten auch die qualitativ besseren Postscript-Schriften zu verwenden. Bevor man die Postscript-Schriften verwenden kann, müssen sie aber zunächst auch in der Win-OS/2-Umgebung installiert werden. Hierzu dient das ATM Control Panel, das in der Programmgruppe *Adobe Type Manager* im Windows-Programm-Manager zu finden ist.

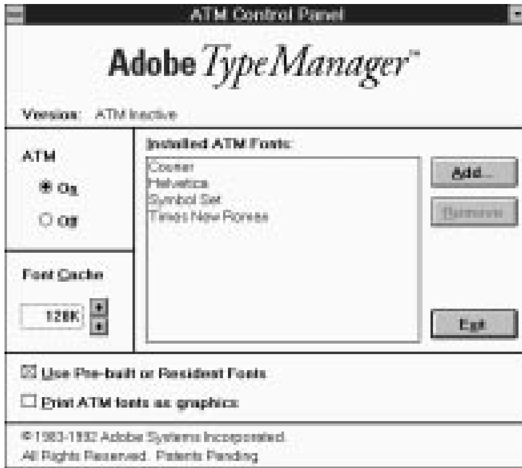


Abb. 8.10: Das ATM Control Panel

Zum Hinzufügen bzw. Entfernen von Schriftarten verwenden Sie die Buttons an der rechten Fensterseite. Links kann die Größe des für Schriftarten verwendeten Puffers eingestellt werden. Je mehr Schriftarten gleichzeitig verwendet werden sollen, desto mehr Speicher sollte dafür vorgesehen werden. Bedenken Sie aber bei der Verwendung von Postscript-Schriftarten, daß diese nicht nur wertvollen Arbeitsspeicher beanspruchen, sondern zudem auch die Ladezeit der Windows-Umgebung beeinflussen. Dies ist auch der Grund, weshalb es die Win-OS/2-Einstellung *WIN_ATM* gibt. Diese Einstellung sollte nur bei solchen Sitzungen auf *On* gesetzt werden, in denen Postscript-Schriftarten verwendet werden. Bei allen anderen Sitzungen sollte die Voreinstellung *Off* belassen werden, damit nicht unnötig Arbeitsspeicher verschwendet wird.

8.6.3 32-Bit-Windows-Programme unter OS/2

Die Win-OS/2-Umgebung unterstützt 32-Bit-Anwendungen, die mit der Win32s-Schnittstelle arbeiten. Während man sich bei OS/2 Warp Version 3 mit der Version 1.15 von Win32s begnügen mußte, unterstützt OS/2 Warp Version 4 die OLE2-fähige Version 1.15, 1.20, 1.25 und 1.25a dieser Schnittstelle. Die Erweiterung der Kompatibilität auf Version 1.30 von Win32s ist geplant. Durch die Unterstützung von Win32s-Anwendungen hat man sogar Zugang zu vielen Anwendungen, die für Windows 95 programmiert wurden. Die meisten dieser Anwendungen verwenden nämlich lediglich die Win32s-Schnittstelle und sind damit auch unter Win-OS/2 ablauffähig.

Um Win32s-Anwendungen unter OS/2 einsetzen zu können, muß zunächst das entsprechende Laufzeitsystem installiert werden. Dieses Laufzeitsystem enthält alle Dateien, die zur Bereitstellung der 32-Bit-Schnittstelle notwendig sind. Es ist in vielen Mailboxen erhältlich und wird häufig sogar bei den Anwendungen, die das Win32s-System benutzen, mitgeliefert.

Sollten Probleme mit einer Win32s-Anwendung auftreten, sollte man zunächst überprüfen, ob die Win32s-Unterstützung installiert ist. Verantwortlich hierfür ist der Treiber vw32s.sys, der in der »config.sys« von OS/2 mit einer Device-Anweisung aktiviert werden muß. Leider gibt es einige Win32s-Programme, die Einrichtungen der Schnittstelle verwenden, die unter OS/2 nicht unterstützt werden (z.B. Adobe Photoshop, EndNote2 Plus und PSpice). Es ist momentan nicht möglich, solche Anwendungen unter OS/2 zu verwenden.

– Für Win-OS/2-Sitzungen, in denen Win32s-Anwendungen ausgeführt werden sollen, sollte die Einstellung *DOS_FILES* auf den Wert 255 gesetzt werden.

– Falls bei Verwendung einer Win32s-Anwendung eine Fehlermeldung angezeigt wird, die besagt, daß zu wenig Speicherplatz für die Anwendung vorhanden ist, verringern Sie den bei *DPMI_MEMORY_LIMIT* eingestellten Wert.

– Zur Ausführung von Visual FoxPro muß die Einstellung *DPMI_MEMORY_LIMIT* auf den Wert 16 gesetzt werden.

Weitere Informationen zur Win32s-Unterstützung unter OS/2 sind in der Online-Dokumentation enthalten. Wichtige Informationen zur Fehlerbehebung finden Sie in der Datei »readme« im Basisverzeichnis des OS/2-Installationslaufwerks.

9 Netzwerkfunktionalität von OS/2 Warp V4

von Florian Piekert, Hendrik Fulda, Oliver Mark und Raimund Mann

9.1 Eine Einführung in TCP/IP

von Florian Piekert und Hendrik Fulda

Im folgenden Kapitel finden Sie alle Informationen, die Sie benötigen, um unter OS/2 Warp Anschluß an das Internet oder ein anderes TCP/IP-Netzwerk zu bekommen. Weiterhin wird die Konfiguration und Funktionsweise der bei OS/2 Warp mitgelieferten TCP/IP-Applikationen beschrieben.

Das Netzwerkprotokoll TCP/IP benötigen Sie, wenn Sie sich an das Internet oder ein TCP/IP-basierendes Firmennetzwerk (Intranet) anschließen wollen.

9.1.1 Was ist TCP/IP?

Wie alle Netzwerkprotokolle ist TCP/IP (Transmission Control Protocol / Internet Protocol) ein Satz von Regeln, der den Austausch von Daten zwischen zwei oder mehr Computersystemen beschreibt. Dazu gehört einmal die Art und Weise, wie die Daten zwischen den Systemen übertragen werden, zum Beispiel die Adressierung und Paketgröße der Daten, aber auch die Art und Weise, wie ein Programm oder ein Programmierer auf das Protokoll zugreifen kann.

TCP/IP wurde in den 70er Jahren in den USA entwickelt und verdankt seinen heutigen Bekanntheitsgrad vor allem der stürmischen Entwicklung des Internet.

Daß es aber überhaupt zu einer solchen Entwicklung kommen konnte, verdanken wir der Tatsache, daß TCP/IP ein offenes Protokoll ist, d.h., die Spezifikationen des TCP/IP-Protokolles liegen öffentlich vor, so daß jeder in der Lage ist, eine TCP/IP-Version selbst zu entwickeln, die mit anderen TCP/IP-Versionen für andere Computersysteme kommunizieren kann.

9.1.2 Eines mit allen, alle mit einem

Es gibt TCP/IP-Versionen für fast jedes Betriebssystem, sei dies nun für PCs, Workstations oder Mainframes. Es gibt TCP/IP für Amiga, Atari und Arcorn Computer, für DOS, alle Windows-Versionen, alle OS/2-Versionen, alle Unices, OS/400, VMS, MVS und vieles mehr.

TCP/IP ermöglicht den plattformübergreifenden Datenaustausch zwischen den unterschiedlichsten Betriebssystemen über die unterschiedlichsten Netzwerkmedien wie Ethernet, Token Ring, Standleitungen, FDDI, ATM etc. TCP/IP ist der kleinste (und manchmal auch der größte) gemeinsame Nenner aller Computersysteme im Internet.

9.1.3 TCP/IP für LANs?

Wenn Sie ein neues Netzwerk aufbauen, haben Sie sich vielleicht gefragt, ob TCP/IP dafür das richtige Protokoll ist.

Die Vorteile von TCP/IP liegen auf der Hand: als plattformübergreifendes Protokoll ist es wie kein anderes für den Einsatz in Umgebungen mit unterschiedlichsten Betriebssystemen prädestiniert. Auch kann man mit TCP/IP sehr große Netzwerke bis bin zu GANs (Global Area Networks) aufbauen, da die Paketverteilung (routing) sehr effektiv gestaltet werden kann. Auch Netzwerkmanagementsysteme, die bei größeren Netzwerken immer notwendiger werden, setzen überwiegend auf TCP/IP auf.

Natürlich hat TCP/IP nicht nur Vorteile. Jedes TCP/IP-Paket enthält die voll qualifizierte Absender- und Empfängeradresse, was zwar für die Paketverteilung von Vorteil ist, letztendlich jedoch dazu führt, daß mehr Daten übertragen werden, als nötig ist. Ein weiterer Nachteil ist das Fehlen jeglicher Sicherheit der Daten vor Lauschern. Bei TCP/IP werden alle Nutzdaten im Klartext übertragen, wenn nicht die Anwendungsprogramme für eine Verschlüsselung sorgen.

Letztendlich kommt es darauf an, welche Dienste durch das LAN zur Verfügung gestellt werden sollen. Wenn ausschließlich Datei- und Druckdienste benötigt werden, nur OS/2, DOS und Windows Arbeitsplätze eingesetzt werden und das LAN nicht über mehrere Standorte ausgedehnt werden soll, reicht das NetBIOS-Protokoll, das von den OS/2-Peer-Netzwerkdiensten, LAN Server und Windows NT verwendet wird, oder das SPX/IPX-Protokoll der Novell-Server vollständig aus.

Wollen Sie jedoch auf Inter- oder Intranetdienste wie World Wide Web (WWW) zugreifen, andere Betriebssysteme wie Unix verwenden, oder ein LAN über mehrere Standorte ausdehnen, ist TCP/IP die richtige Wahl.

Außerdem haben Sie immer die Möglichkeit, auch weitere Netzwerkprotokolle parallel zu installieren. OS/2 Warp und Ihre Netzwerkhardware, daß heißt, die Kabel und die Netzwerkkarten, sind in der Lage, mehrere Netzwerkprotokolle parallel zu übertragen, man kann also auf einem Ethernet oder Token-Ring-Netzwerk gleichzeitig TCP/IP- und beispielsweise das SPX/IPX-Protokoll benutzen. Verschiedene Computersysteme können sich dann sowohl über TCP/IP als auch über SPX/IPX miteinander unterhalten, ohne daß sich die Datenströme gegenseitig stören.

In diesem Buch gehen wir davon aus, daß Sie einen oder mehrere PCs unter OS/2 Warp an das Internet anschließen möchten. Die Herangehensweise unterscheidet sich jedoch nicht von einem Anschluß an ein »einfaches« TCP/IP-LAN – Sie werden vielleicht nur nicht so viele Dienste verwenden können.

9.2 Grundlagen I: Die Adressen bei TCP/IP

von Florian Piekert und Hendrik Fulda

9.2.1 Wofür Adressen?

Um Daten über ein Netzwerk übertragen zu können, werden diese in einzelne Datenpakete aufgeteilt, die eine bestimmte Länge haben. Um die einzelnen Datenpakete korrekt zuordnen und weiterleiten zu können, werden sie mit einer Empfänger- und Absenderadresse versehen.

Adressiert wird jedoch nicht das gesamte Computersystem, sondern jede Netzwerkschnittstelle (Interface) eines Systems für sich. Egal ob die Netzwerkschnittstelle eine Netzwerkkarte oder eine serielle Schnittstelle ist. Der »kleinste« Anschluß, über den man TCP/IP fahren kann, ist die serielle Schnittstelle eines Computersystems. Alternativ oder zusätzlich werden Netzwerkkarten verwendet. Die meisten Computersysteme haben jedoch nur eine Netzwerkschnittstelle.

9.2.2 Die IP-Adresse

Eine TCP/IP-Adresse besteht aus einem 32-Bit-Wert, der durch vier durch Punkte getrennte Zahlen dargestellt wird. Diese Zahlen sind jeweils ein Byte groß, d.h. sie können die (dezimalen) Werte 0 bis 255 annehmen.

Eine korrekte TCP/IP-Adresse wäre zum Beispiel: 208.5.0.7

Jede IP-Adresse darf nur einmal in einem Netzwerk vorkommen. Koppelt man mehrere Netzwerke zusammen – so wie das im Internet der Fall ist – gilt dies um so mehr.

Für die Adressierung eines fernen TCP/IP-Systems müssen wir mindestens über diese Adresse verfügen.

9.2.3 Der IP-Name

Weil aber die auf Nummern basierenden IP-Adressen nicht leicht zu merken sind, hat man die Möglichkeit geschaffen, statt IP-Nummern sogenannte »sprechende« oder symbolische Adressen zu verwenden, die dann zum Beispiel »www.teamos2.de« lautet.

Die Umsetzung von IP-Nummern zu symbolischen IP-Namen geschieht durch Tabellen, die entweder auf dem einzelnen Computersystem bereitgehalten werden oder – und das ist der Normalfall – durch einen Server, der die Tabellen für ein ganzes Netz verwaltet.

Dieser Dienst heißt »Domain Name Service« (DNS). Wenn Sie ein Computersystem also über einen IP-Namen ansprechen, wird immer erst der DNS-Server zu Rate gezogen, weil die Datenpakete nur mit IP-Nummern versehen verschickt werden können.

Um diesen Dienst und damit die leicht zu merkenden IP-Namen verwenden zu können muß uns nur einmal die IP-Nummer des DNS-Servers bekannt sein, damit weitere Zuordnungen erfragt werden können.

9.2.4 Der Domain Name Service – IP-Namen und ihr Aufbau

Da Sie im Internet fast ausschließlich über IP-Namen navigieren werden, lassen Sie uns einen genaueren Blick auf den Aufbau von IP-Namen und dem Domain-Name-Service werfen.

IP-Namen sind hierarchisch aufgebaut und werden von hinten nach vorne gelesen. Eine voll qualifizierte symbolische IP-Adresse besteht immer aus einer Toplevel-Domain, einer Haupt-Domain und einem Systemnamen. Die Adresse »www.teamos2.de« bezieht sich also auf das Computersystem »www« in der Haupt-Domain »teamos2« aus der Toplevel-Domain »de«.

Innerhalb jeder Hierachiestufe darf ein Name immer nur einmal vorkommen. Es gibt also nur eine Toplevel-Domain »de«, darunter nur eine Haupt-Domain mit dem Namen »teamos2« und in dieser nur ein System mit dem Namen »www«. Auf jeder Hierachiestufe können beliebig viele Namen eingerichtet werden.

Die Toplevel-Domain-Namen sind im Internet weltweit festgelegt. Um die Namensvergebung zu verstehen, muß man sich vor Augen halten, daß das Internet in den USA »erfunden« wurde und man am Anfang wohl gar nicht an einen weltweiten Ausbau gedacht hat. So gibt es zum Beispiel folgende Toplevel-Domain-Namen, die zwar eigentlich länderübergreifend gelten könnten, sich aber nur auf die USA beziehen:

– .gov für Regierungssysteme

– .mil für Computer des Militärs

– .edu für Universitäten

Andere Toplevel-Domain-Namen dürfen weltweit benutzt werden:

– .org für Organisationen

– .com für kommerzielle Anbieter

Weiterhin gibt es keine Toplevel-Domain für die USA, sondern jeweils eine für jeder Staat, zum Beispiel ».ny« für New York.

Für jedes andere Land gibt es nur eine Toplevel-Domain, beispielsweise »de« für Deutschland und »fi« für Finnland.

Die weltweiten Toplevel-Domain-Namen und die der USA werden durch das »Internet Network Information Center« (InterNIC – www.internic.net) verwaltet – die Toplevel-Domain-Namen der anderen Länder durch NICs in den jeweiligen Ländern – in Deutschland durch das DE-NIC (www.nic.de).

Das DE-NIC ist also für alle Domain-Namen unterhalb von ».de« zuständig. Dorthin müssen Sie also Ihre Anfrage für die Vergabe eines eigenen Domain-Namens richten. Im Normalfall macht dies aber Ihr Internet-Service-Provider (ISP) für Sie. Wenn Sie eine eigene Sub-Domain

besitzen, können Sie unterhalb dieser Sub-Domain weitere Domänen einrichten oder aber Computersysteme einrichten.

Verschachtelungen wie »notes.hamburg.teamos2.de« oder »notes.blankenese.hamburg.teamos2.de« sind durchaus möglich – Grenzen sind hier nur durch Ihre Phantasie gesetzt. Und um noch einen draufzusetzen: Ein Computersystem kann auch unter mehreren Namen bekannt sein – »www.hamburg.teamos2.de« und »notes.hamburg.teamos2.de« können also durchaus dasselbe Computersystem sein.

Die NICs vergeben auch die »echten« IP-Adressen. Da diese im gesamten Internet auch immer nur einmal vorkommen dürfen, ist eine solche zentrale Verwaltung notwendig. Da auch langsam der Adreßraum knapp wird, kann man als Firma im Normalfall keine eigenen Adressen mehr beantragen, sondern bekommt diese durch seinen ISP zugewiesen.

9.2.5 Und in der Praxis?

In diesem Buch werden wir die Begriffe »IP-Adresse« oder »IP-Nummer« in Zukunft für die »echte«, numerische IP-Adresse verwenden und den Begriff »IP-Name« für symbolische IP-Adressen.

Wenn bei der Konfiguration von TCP/IP oder bei der Benutzung von TCP/IP-Programmen die Adresse eines Computersystemes angegeben werden muß, kann überall bis auf wenige Ausnahmen sowohl der IP-Name als auch die IP-Nummer angegeben werden.

9.3 Grundlagen II: Aufbau von IP-Netzen

von Florian Piekert und Hendrik Fulda

9.3.1 Von großen und kleinen Netzen

Beim Aufbau von großen Netzwerken oder dem Zusammenschluß von mehreren Netzwerken reicht es nicht aus, nur für jedes Computersystem eine Adresse zu besitzen. Man muß die Möglichkeit haben, mehrere Computersysteme logisch zusammenzufassen, um so Datenströme sinnvoll kanalisieren zu können.

Stellen Sie sich vor, Sie möchten zwei Netzwerke über eine Telefonleitung verbinden. Um zu verhindern, daß Daten, die nicht für das jeweils andere LAN bestimmt sind, übertragen werden und so zu Engpässen führen, wäre es sinnvoll, schon an der Adresse die Zugehörigkeit eines Systems zu einem Netz feststellen zu können.

Wenn man jedes System mit einer fortlaufenden Adresse versehen würde, müßte man in den einzelnen Verbindungssystemen (Router) riesige Tabellen erstellen, die festhalten, welches System wo zu erreichen ist, da in diesem Fall ein Eintrag pro System notwendig wäre.

Um also eine logische Zusammenfassung von Computersystemen zu ermöglichen, ist eine IP-Adresse noch einmal unterteilt – und zwar in eine Netz- und eine System-Adresse.

Die Zahlen der System-Adresse bezeichnen jeweils ein bestimmtes Computersystem aus einem durch die Netz-Adresse bestimmten Netz.

9.3.2 Die IP-Netzwerkmaske

Bei TCP/IP erfolgt die Aufteilung der IP-Adresse in System- und Netz-Adresse nicht statisch, sondern wird durch eine sogenannte Netzwerkmaske (Netmask) oder TeilNetzwerkmaske (Subnet-Mask) festgelegt.

Die Netzwerkmaske beschreibt direkt, welche Bit der IP-Adresse zur Netz-Adressierung verwendet werden und ist auch genau wie eine IP-Adresse aufgebaut. Die »verbleibenden« Bit der IP-Adresse stehen für die Adressierung der Computersysteme zur Verfügung.

Die Netz-»Maskierung« findet auf Bit-Ebene statt. Jedes Bit der Netzwerkmaske ist dem äquivalenten Bit der IP-Adresse zugeordnet. Bei der Netzwerkmaske wird nun jedes Bit der IP-Adresse, das zur Netz-Adresse gehört, auf Eins gesetzt.

Beispiel: Bei einer IP-Adresse 123.123.123.123 mit einer Netzwerkmaske von 255.255.255.0 sind die ersten drei Byte vollständig maskiert, es steht also nur das letzte Byte zur Adressierung von Computersystemen zur Verfügung.

Es können also 256 Systeme adressiert werden.

Lautete die Netzwerkmaske jedoch 255.255.0.0, stünden zwei Byte zur Verfügung und es könnten 256 mal 256, also 65536 Systeme, adressiert werden.

9.3.3 Und in der Praxis?

Um ein fremdes Computersystem adressieren zu können, muß uns die Netzwerkmaske seiner IP-Adresse nicht bekannt sein. Jedes Computersystem muß jedoch wissen, wie es seine eigene IP-Adresse aufzuteilen hat und ist deshalb auf die Angabe einer Netzwerkmaske angewiesen.

Da IP-Adressen für jede Netzwerkschnittstelle vergeben werden, muß auch für jede Netzwerkschnittstelle eine Netzwerkmaske angegeben werden.

Weiterhin müssen die Netzwerkmasken für alle unmittelbar über die eigene Netzwerkschnittstelle erreichbaren Netze bekannt sein.

9.4 Grundlagen III: Routing

von Florian Piekert und Hendrik Fulda

Wenn Sie nur ein einziges Netz betreiben, ist das Festlegen von Verbindungsleitwegen (Routen) zu anderen Netzwerken müßig. Sobald Sie aber über einen oder mehr Anschlüsse an andere Netze angeschlossen sind, muß festgelegt werden, welche Computersysteme als Verbindungssysteme zu welchen Netzen fungieren. Dies wird als »Routing« bezeichnet. Es werden sogenannte »Routingtabellen« angelegt, die die angeschlossenen Netze beschreiben.

Es gibt drei verschiedene Arten von Routen. Einmal Routen zu ganzen Netzen, Routen zu Teilnetzwerken (Subnetzen) und Routen zu einzelnen Computersystemen (Hosts). Weiterhin wird meistens auch noch eine Standard-Route (default route) vergeben, die das Verbindungssystem (Router) zu allen nicht speziell definierten Netzen und Computersystemen angibt.

Für das einzelne Computersystem ist das Anlegen einer Routingtabelle gewöhnlich trivial: Es wird nur eine Route für das lokale Netz definiert und falls Wege zu anderen Netzen existieren, wird noch eine Route zu dem zuständigen Vermittlungsrechner (Router) angegeben – fertig.

Erst wenn Sie selbst einen Vermittlungsrechner einrichten, um beispielsweise ein LAN an das Internet anzuschließen, wird es etwas komplizierter.

In der Praxis werden Sie überwiegend auf drei verschiedene Routing-Szenarien treffen. Dies ist einmal die Verwendung Ihres Computersystems in einem LAN, über eine Modemverbindung oder als Verbindungssystem zwischen einem LAN und dem Internet über ein Modem.

9.5 Praxis I: Installation

von Florian Piekert und Hendrik Fulda

Vor der Installation von OS/2 Warp sollten Sie klären, wie Sie von OS/2 auf ein TCP/IP-Netz zugreifen wollen. Es gibt drei Möglichkeiten: Über ein Modem, über ein LAN oder gleichzeitig über Modem und LAN. Letzteres kann sinnvoll sein, wenn Sie an Ihrem Arbeitsplatz oder zu Hause ein TCP/IP-LAN haben, aber zusätzlich noch über Modem auf das Internet zugreifen wollen oder sogar Ihr LAN ans Internet anschließen möchten.

9.5.1 TCP/IP bei der Installation von OS/2 Warp

Wenn Sie TCP/IP auch über ein LAN benutzen wollen, müssen Sie bei der Installation von OS/2 Warp die *TCP/IP Dienste* auswählen.



Abb. 9.1: Die OS/2-Warp-Anpassung und Installation

Als nächstes müssen Sie die TCP/IP-Dienste konfigurieren. Dazu wird erst der Konfigurationsbildschirm für die Netzwerkdienste angezeigt, in dem Sie dann die einzelnen Dienste auswählen und konfigurieren.

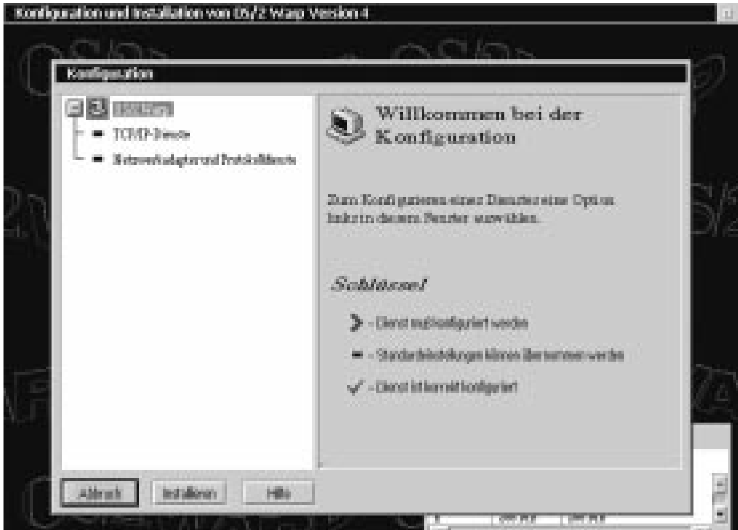


Abb. 9.2: Die Konfiguration der Netzwerkdienste

9.5.2 Die TCP/IP-Dienste (LAN) konfigurieren

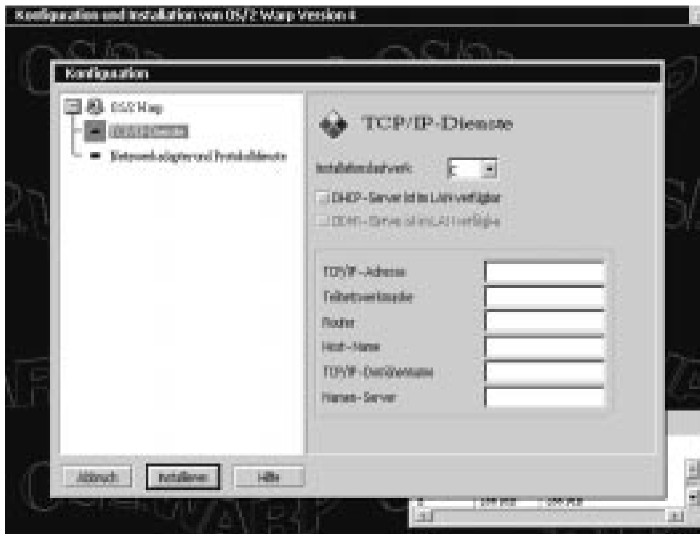


Abb. 9.3: Der TCP/IP-Dienste-Konfigurationsbildschirm

Installationslaufwerk

Geben Sie das Laufwerk an, auf dem die zu TCP/IP gehörenden Dateien abgelegt werden sollen. Es ist nicht notwendig, TCP/IP auf dem OS/2-Warp-Startlaufwerk zu installieren.

DHCP Server ist im LAN verfügbar

Wenn Sie in einem TCP/IP-Netzwerk arbeiten, in dem ein DHCP-Server installiert ist, dann können Sie sich das Eintragen der TCP/IP-Adresse, Netzwerkmaske etc. sparen. DHCP steht für »Dynamic Host Configuration Protocol« und ist ein offener Standard zur zentralen Verwaltung von TCP/IP-Adressen durch einen DHCP-Server und die Übergabe dieser Adressen an die Netzwerk-Arbeitsplätze. Kurz: Die Verwendung eines DHCP-Servers entbindet den Netzwerkverwalter oder den Benutzer von der Aufgabe, TCP/IP-Adressen am Arbeitsplatz zu konfigurieren.

Der DHCP-Client ist jetzt Bestandteil von OS/2 Warp. Ein DHCP-Server ist in der IBM-Warp-Server-Version-4 enthalten, es können aber auch andere DHCP-Server wie verschiedene Unices oder Windows-NT-Server verwendet werden.

DDNS Server ist im LAN verfügbar

Bei der zusätzlichen Verwendung eines DDNS-Servers kann sich DHCP vollständig entfalten. DDNS steht für »Dynamic Domain Name Services« und ist eine Erweiterung des klassischen Domain-Name-Service- (DNS) Dienstes um die Möglichkeit, die DNS-Tabellen dynamisch zu aktualisieren, wenn eine Arbeitsstation (z.B. ein Laptop) seinen Standort im TCP/IP-LAN oder -WAN verändert. Ein DDNS-Server ist im Warp Server enthalten.

TCP/IP-Adresse

Geben Sie hier die IP-Adresse ein, die Sie vom Netzadministrator Ihres Netzes erhalten haben, oder die Sie für diesen Arbeitsplatz vorgesehen haben. Zum Beispiel 192.168.0.1.

Teilnetzwerkmaske

Geben Sie hier die Teilnetzwerkmaske für die IP-Adresse ein. Für das obige Beispiel wäre das 255.255.255.0.

Router

Hier geben Sie die IP-Adressen des Standard-Routers für Ihr LAN ein. Dieser Eintrag wird manchmal auch als »default Router« oder »default Gateway« bezeichnet. Wenn Sie keinen Router für den Übergang in andere Netze haben, oder wenn dieses Computersystem der Router werden soll, tragen Sie hier auch die IP-Adresse Ihres eigenen Computers ein.

Host-Name

Hier wird der IP-Name dieses Computersystems eingetragen – und zwar ohne die Domain. Zum Beispiel also »merlin«.

TCP/IP-Domänenname

An dieser Stelle wird der Domain- und Toplevel-Domain-Name des TCP/IP-Netzwerkes eingetragen. Zum Beispiel »teamos2.de«.

Namen-Server

Wenn das TCP/IP-Netzwerk über einen DNS-Server verfügt, muß dessen **numerische** IP-Adresse hier angegeben werden, damit IP-Namen zur Adressierung von Computersystemen benutzt werden können. Wenn Ihr Netzwerk über keinen DNS verfügt, lassen Sie dieses Feld frei.

9.5.3 Das TCP/IP-Protokoll hinzufügen

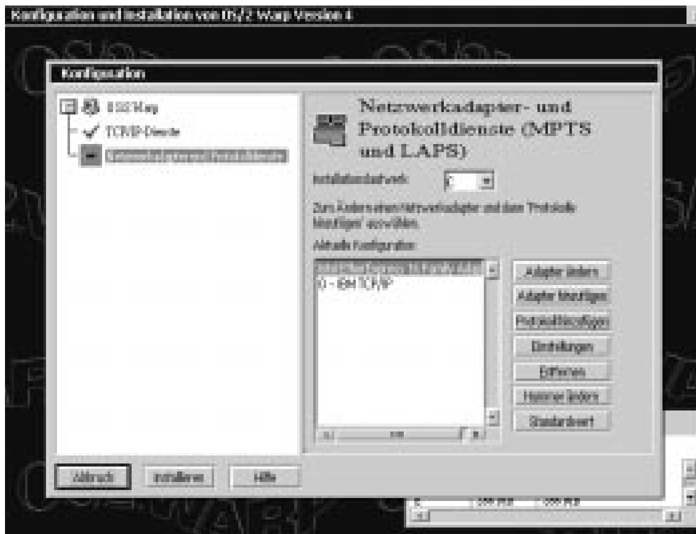


Abb. 9.4: Netzwerkadapter und Protokolldienste konfigurieren

Achten Sie darauf, daß hier das Protokoll »IBM TCP/IP« unter der aktuellen Konfiguration angezeigt wird. Sollte dies nicht der Fall sein, fügen Sie das TCP/IP Protokoll über *Hinzufügen* hinzu.

Damit ist der TCP/IP-spezifische Teil der Installation abgeschlossen.

9.6 TCP/IP nach der Installation von OS/2 Warp

von Florian Piekert und Hendrik Fulda

Zur nachträglichen Änderung der TCP/IP-Einstellungen und aller Parameter, die bei der Installation nicht abgefragt wurden, gibt es das *TCP/IP Konfiguration (LAN)*-Einstellungsnotizbuch. Hier sind beim fertigen OS/2 Warp die Werte aus der Installationsprozedur zu finden. Im folgenden wollen wir die einzelnen Seiten näher betrachten.

Wenn Sie OS/2 Warp neu installiert haben, wollen Sie unter Umständen folgende Einstellungen ergänzen (eine Beschreibung der Einträge finden Sie auf den nächsten Seiten):

– *Benutzername* im Abschnitt *Allgemein*

– *NewsReader/2 News Server* im Abschnitt *Server*

– *Post-Benutzer-ID* unter dem Abschnitt *Post*, Seite 1, ändern

– *POP-Post-Server, POP ID* und *POP-Kennwort* unter dem Abschnitt *Post*, Seite 2, eintragen

– *Antwortdomäne* (Reply-Domain) und *SMTP-Post-Gateway* unter dem Abschnitt *Sendmail*, Seite 1

Um die Einstellungen vorzunehmen, öffnen Sie den Ordner *Programme* auf der Arbeitsoberfläche, dann den Ordner *Internet (LAN)* und dort das *TCP/IP-Konfiguration (LAN)*-Notizbuch.

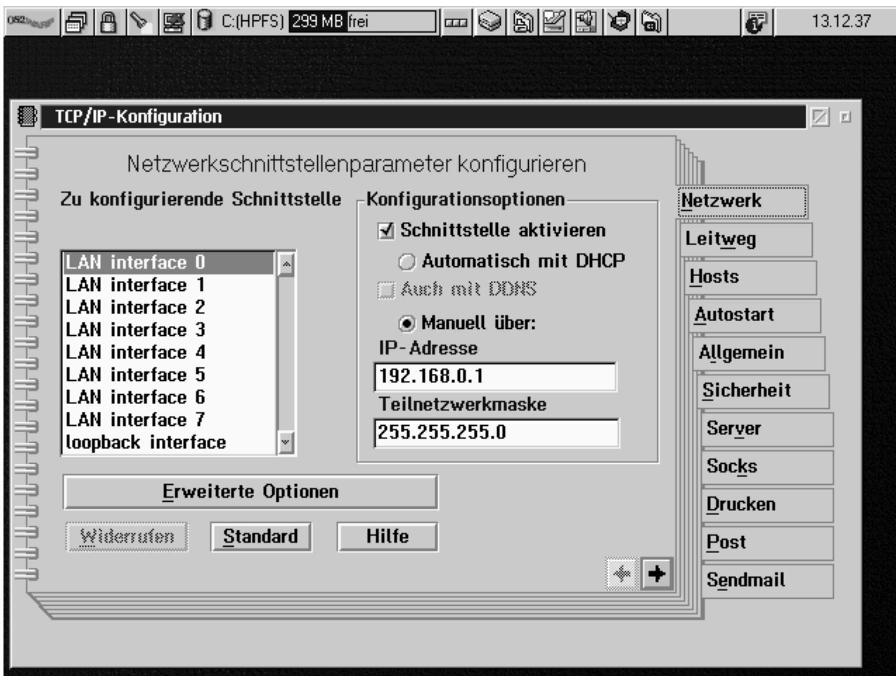


Abb. 9.5: TCP/IP-Konfiguration, Abschnitt Netzwerk

Abschnitt 1: Netzwerk, Netzwerkschnittstellenparameter konfigurieren

Hier konfigurieren Sie jede Ihrer Netzwerkschnittstellen (LAN Interface). Es können bis zu acht LAN-Schnittstellen eingesetzt werden. Für jede Netzwerkschnittstelle müssen Sie eine IP-Adresse und die Teilnetzwerkmaske für diese Adresse angeben. In den erweiterten Optionen kann eine abweichende Rundsendeadresse (Broadcast Adress) angegeben werden. Auch Einstellungen für komplexere Routen (Knotenanzahl / Metric counts) und Punkt-zu-Punkt-Verbindungen werden hier für jede Netzwerkschnittstelle definiert.

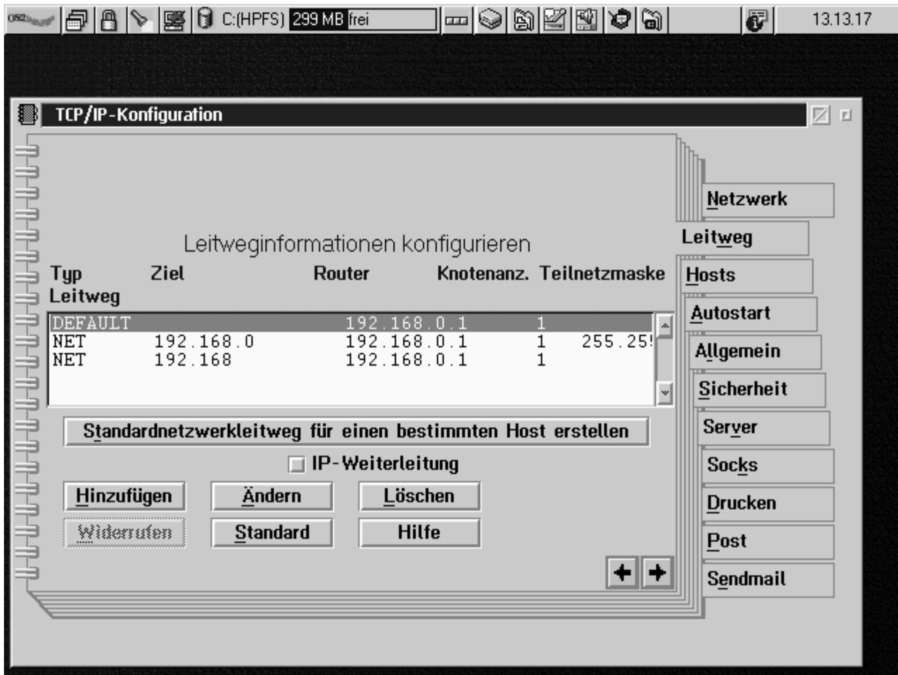


Abb. 9.6: TCP/IP-Konfiguration, Abschnitt Leitweg

Abschnitt 2: Leitweg, Leitweginformationen konfigurieren

Hier werden alle Leitwege (Routen) definiert. Es muß mindestens eine Route für das lokale LAN definiert sein. Diese ist als Routen-Typ *Netz* einzutragen.

Wenn Sie schon bei der Installation von OS/2 Warp eine eigene TCP/IP-Adresse für Ihr Computersystem angegeben haben, sind hier schon zwei Einträge gesetzt, die sich auf Ihre Konfiguration beziehen.

Über die Funktionen *Hinzufügen*, *Ändern* und *Löschen* können Sie hier bestehende Leitwege ändern oder löschen und neue Leitwege hinzufügen. Wenn Sie mehr als eine Netzwerkschnittstelle haben und/oder Ihr Computersystem als IP-Router zu anderen Netzen verwendet wird, müssen Sie die Checkbox *IP-Weiterleitung* mit einem Haken versehen. Es werden dann IP-Pakete, die von einer Netzwerkschnittstelle kommen, auf andere umgeleitet, wenn entsprechende Leitwege gesetzt sind.



Abb. 9.7: TCP/IP-Konfiguration, Abschnitt Hosts, Seite 1

Abschnitt 3: Hosts Seite 1: Dienste zur LAN-Namensauflösung konfigurieren

Host-Name dieser Maschine

Dieser Parameter ist unter Umständen schon gesetzt. Hier wird der IP-Name dieses Computersystems eingetragen – und zwar ohne die Domain. Zum Beispiel also »merlin«. Diesen Eintrag finden Sie in der CONFIG.SYS als »SET HOSTNAME=<Host-Name>«

Name der lokalen Domäne

Dieser Parameter ist unter Umständen schon gesetzt. An dieser Stelle wird der Domain- und Toplevel-Domain-Name des TCP/IP-Netzwerkes eingetragen. Zum Beispiel »teamos2.de«. Diesen Eintrag finden Sie in der ersten Zeile der Datei \MPTN\ETC\RESOLV2 nach dem Schlüsselwort »domain«.

LAN-Namen-Server-Konfiguration

Namen-Server-Adressen

Dieser Parameter ist u.U. schon gesetzt. Wenn Ihr TCP/IP-Netzwerk über einen DNS-Server verfügt, muß dessen **numerische** IP-Adresse hier angegeben werden, damit IP-Namen zur Adressierung von Computersystemen benutzt werden können. Wenn Sie über keinen DNS verfügen, lassen Sie dieses Feld frei. Es können bis zu drei DNS-Server eingetragen werden.

Diese werden dann der Reihe nach bei der Übersetzung von IP-Namen in IP-Adressen abgefragt. Diese Einträge finden Sie in der Datei \MPTN\ETC\RESOLV2\ nach dem Schlüsselwort »nameserver«.

LAN Domänensuchliste

Wenn Sie ein Computersystem ansprechen wollen, müssen Sie immer den vollen IP-Namen mit Domain verwenden. Wenn Sie allerdings ein Computersystem innerhalb Ihrer eigenen Domain ansprechen, können Sie auch nur den Host-Namen verwenden, da Ihr eigener Domain-Name angehängt wird. Arbeiten Sie normalerweise mit Computersystemen aus bestimmten Domains, können Sie diese hier eintragen. Dann wird ein Computersystem, das Sie nur über den Host-Namen angesprochen haben, in allen Domains gesucht, die hier angegeben sind.

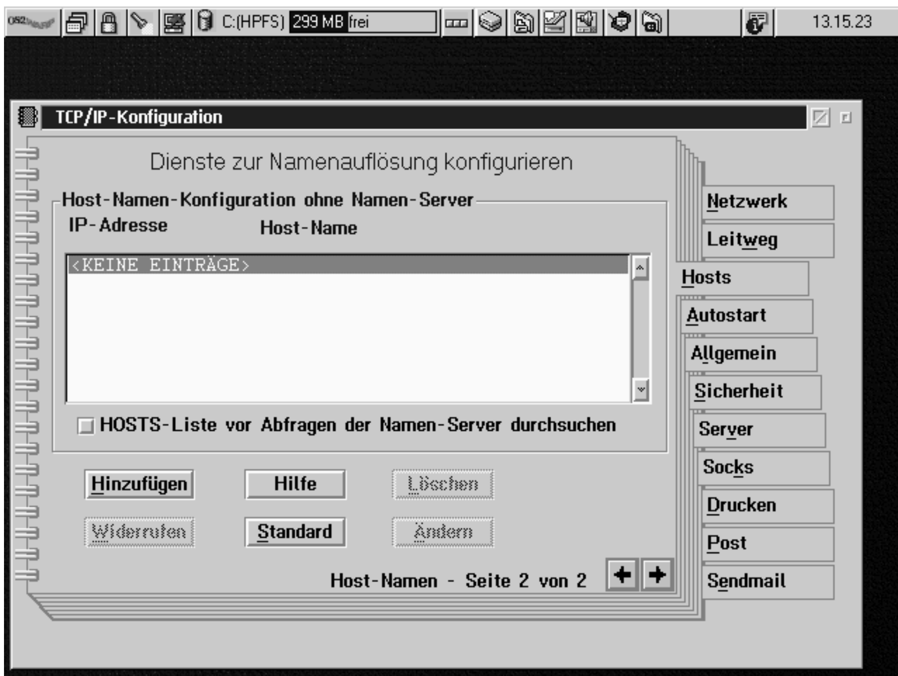


Abb. 9.8: TCP/IP-Konfiguration, Abschnitt Hosts, Seite 2

Seite 2:

Host-Namen-Konfiguration ohne Namen-Server

Wenn Sie keinen DNS-Server verwenden, können Sie trotzdem mit IP-Namen arbeiten. Sie müssen dann allerdings auf jedem Computersystem eine Liste der Host-Namen und der dazugehörigen IP-Adressen verwalten. Hier tragen Sie die IP-Adressen und die dazugehörigen Namen ein. Sie können für ein Computersystem auch mehrere Namen angeben. Diese soge-

nannten Aliase können dann auch verwendet werden. Dies kann sehr hilfreich sein, wenn Sie häufig mit bestimmten Computersystemen arbeiten, die sehr lange IP-Namen haben. Für diese können Sie sich leicht zu merkende Aliase ausdenken und für sich hier konfigurieren.

Um einen neuen IP-Namen oder ein Alias einzutragen, klicken Sie auf Hinzufügen und geben dann die IP-Adresse, den Host-Namen (mit oder ohne Domain) und, optional, ein Aliasnamen ein. Auch ein kurzer Kommentar kann einem Eintrag hinzugefügt werden.

Diese Einträge finden Sie später in der Datei \MPTN\ETC\HOSTS.

Wenn diese Liste (die HOSTS-Datei) abgefragt werden soll, bevor der DNS-Server zu Rate gezogen wird, versehen Sie das Feld HOSTS-Liste vor Abfragen der Namen-Server durchsuchen mit einem Haken. Es wird dann der Eintrag »SET USE_HOSTS_FIRST=1« der CONFIG.SYS hinzugefügt.

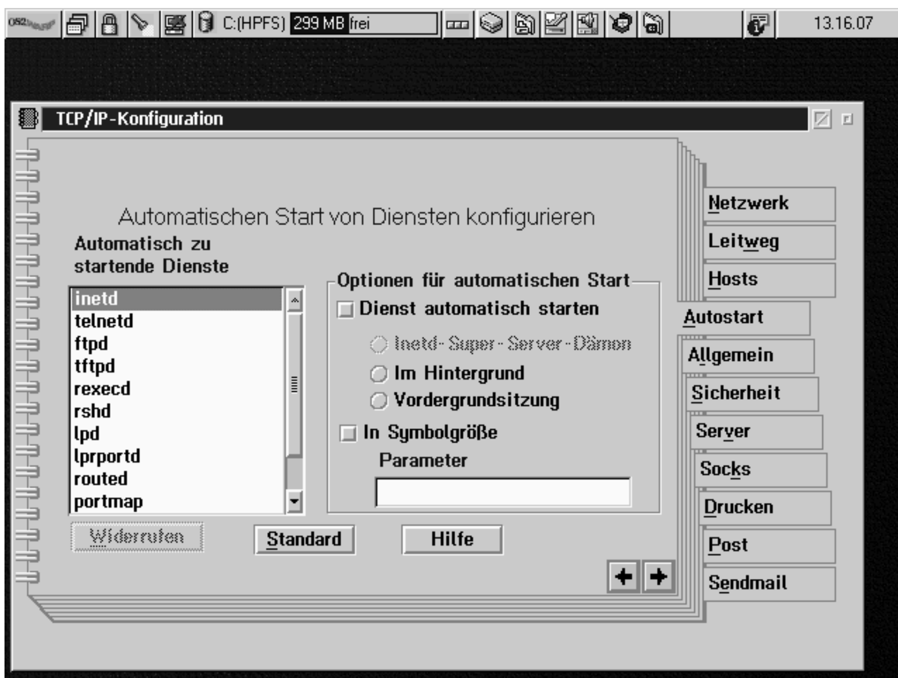


Abb. 9.9: TCP/IP-Konfiguration, Abschnitt Autostart

Abschnitt 4: Autostart, automatisch zu startende Dienste

OS/2 Warp kommt mit verschiedenen TCP/IP-Dienst- und Serverprogrammen. Diese Dienste werden in der Unix-Welt als »Dämonen« bezeichnet. Es sind Hintergrundprozesse, die meistens beim Systemstart aufgerufen werden und dann anderen Benutzern oder Hosts erlauben, Dienste auf diesem System in Anspruch zu nehmen. Teilweise werden diese »Dämonen« auch benötigt, um Dienste auf anderen Computersystemen in Anspruch zu nehmen.

Jeder dieser Dienste wird separat konfiguriert und gestartet. Sie sollten nur die Dienste starten, die Sie auch benötigen, um Systemressourcen nicht unnötig zu binden.

Die einzelnen Dienste sind weiter hinten näher beschrieben. Hier wollen wir auf die Startmöglichkeiten näher eingehen.

Sie können die meisten Dienste entweder im Hintergrund (DETACHED), als Vordergrundsitzung oder über den inetd starten.

inetd – Der inetd ist eine Art »Superdämon«, der Verbindungsanfragen für alle ihm untergeordneten Dienste erkennt und dann den zugehörigen Dienst startet. So werden nur die »Dämonen« gestartet, die auch benötigt werden.

Im Hintergrund – Dienste, die »im Hintergrund« gestartet werden, werden über den Befehl DETACH aufgerufen und erscheinen nicht in der OS/2-Fensterliste.

Vordergrundsitzung – Dienste, die als Vordergrundsitzung aufgerufen werden, starten als ganz normale OS/2-Anwendung und Sie können die Ausgaben des Dienstes am Bildschirm mitverfolgen. Dies ist beispielsweise für den Talk-Dämon notwendig, damit Sie feststellen können, wer versucht Sie zu erreichen.

Weiterhin können Sie für jeden Dienst Parameter übergeben. Dies ist zum Beispiel bei »Sendmail« notwendig.

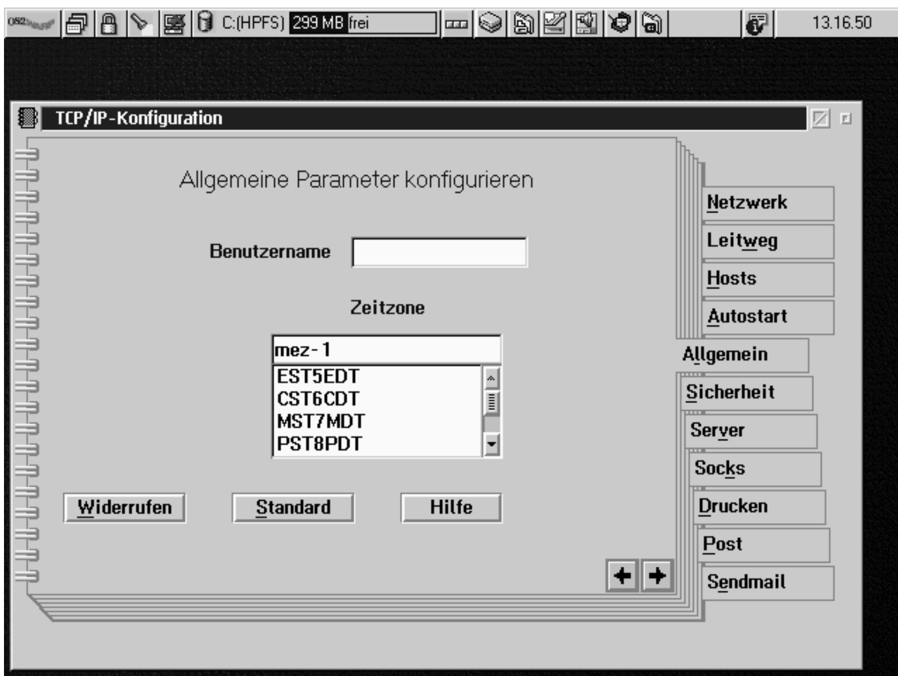


Abb. 9.10: TCP/IP-Konfiguration, Abschnitt Allgemein

Abschnitt 5: Allgemein, allgemeine Parameter konfigurieren

Benutzername

Den Namen oder das Kürzel des an diesem Computersystem arbeitenden Menschen. Der Benutzername wird dann in der »config.sys« mit »SET USER=<Benutzername>« gesetzt und von verschiedenen TCP/IP-Dienstprogrammen und Anwendungen, wie zum Beispiel »Rexec« und »Talk«, ausgelesen.

Zeitzone

Hier ist die Zeitzone einzutragen, in der sich Ihr Computersystem befindet. Für Deutschland ist dies MEZ-1 als Voreinstellung. Dieser Parameter wird in der »config.sys« mit der Umgebungsvariable »SET TZ=<Zeitzone>« gesetzt.



Abb. 9.11: TCP/IP-Konfiguration, Abschnitt Sicherheit, Seite 1

Abschnitt 6: Sicherheit, Seite 1: Server-Sicherheit konfigurieren

Telnet-Kennwort

Hier wird das Kennwort für die Benutzung des Telnet-Dienstes von anderen Computersystemen gesetzt. Diesen Parameter finden Sie leider in der »config.sys« als »SET TELNET.PASSWORD.ID=<Kennwort>« wieder. Siehe auch die Beschreibung des TelnetD-Dienstprogramms.

ftp-Zugriffsschutz (Trusers)

Bei der Benutzung von ftp müssen sich Benutzer von anderen Hosts mit einem Benutzernamen und Kennwort anmelden. So ist es möglich, jedem Benutzer bestimmte Rechte zuzuteilen. So kann man zum Beispiel einstellen, daß der Benutzer nur auf bestimmte Verzeichnisse oder Laufwerke zugreifen darf.

Mit *Hinzufügen* richten Sie einen neuen Benutzer ein. Alle Einträge werden in der Datei \mptn\etc\trusers abgelegt. Siehe auch die Beschreibung des ftpd-Dienstprogramms.



Abb. 9.12: TCP/IP-Konfiguration, Abschnitt Sicherheit, Seite 2

Seite 2:

Kennwort für Rexec-Benutzer

Das Kennwort für die Benutzung von Rexec von entfernten Computersystemen muß hier eingetragen werden. Es wird mit »SET PASSWD=<Kennwort>« in der »config.sys« eingetragen.

Hosts mit Berechtigung für RSH-Server (RHOSTS)

Wenn Benutzer von anderen Hosts auf Ihrem Computersystem den RSH-Dienst verwenden wollen, müssen sie zuvor freigeschaltet werden. Über *Hinzufügen* müssen Sie jeden Benutzer und dessen Computersystem einzeln definieren. Siehe auch die Beschreibung des RSH-Dienstprogramms. Die zugehörige Konfigurationsdatei ist \MPTN\ETC\RHOSTS.

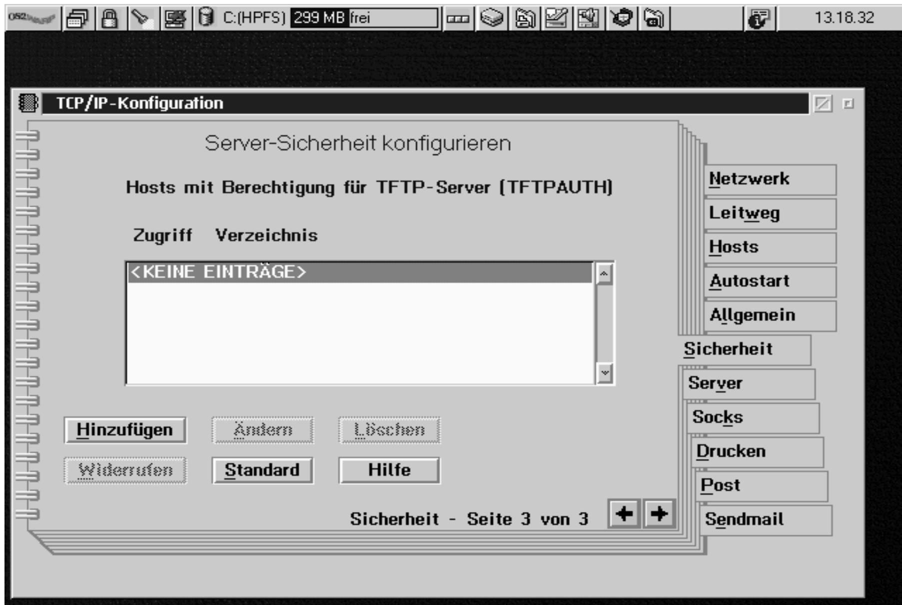


Abb. 9.13: TCP/IP-Konfiguration, Abschnitt Sicherheit, Seite 3

Seite 3:

Hosts mit Berechtigung für tftp-Server (tftpauth)

Verbindungen über tftp werden direkt und ohne Kennwortabfrage entgegengenommen. Hier können allerdings Rechte für einzelne Hosts vergeben werden. Über *Hinzufügen* können Sie einzelnen Hosts Lese- oder Schreib-/Lesezugriffe erlauben. Siehe auch die Beschreibung des tftp-Dienstprogramms. Die zugehörige Konfigurationsdatei ist \mptn\etc\tftpauth.

Abschnitt 7: Server, Server für Anwendungen konfigurieren

Hier werden die Standard-Server für News (NewsReader/2-Server), Gopher (Gopher-Server) und WWW (WWW-Server) definiert. Sie können diese aber auch in den jeweiligen Anwendungen einstellen.

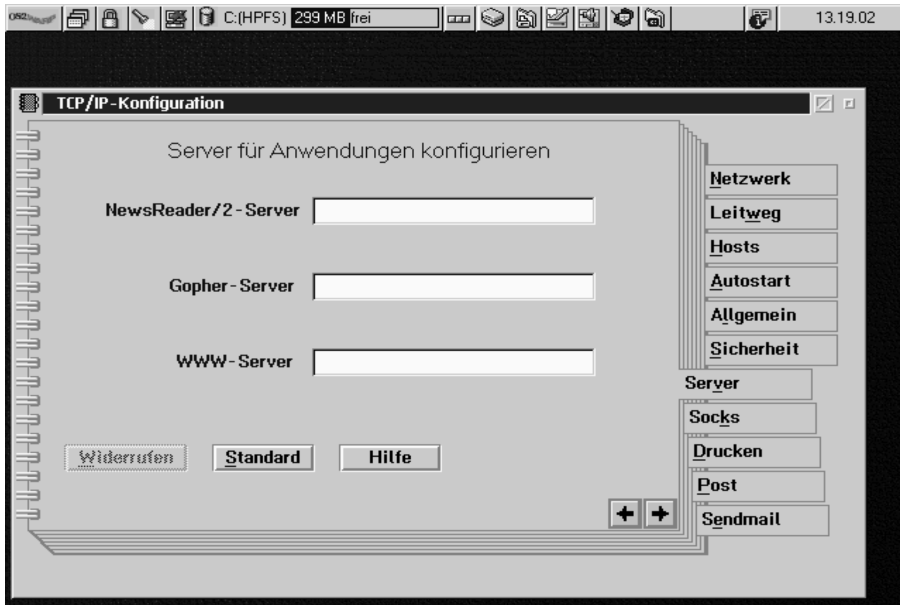


Abb. 9.14: TCP/IP-Konfiguration, Abschnitt Server



Abb. 9.15: TCP/IP-Konfiguration, Abschnitt Socks, Seite 1

Abschnitt 8: Socks, Seite 1: Socks-Standardwerte konfigurieren

»Socks« ist eine »Firewall«-Technologie. Anstatt direkt vom eigenen Computersystem eine Verbindung zum Zielsystem aufzubauen, wird sämtliche Kommunikation in externe Netze über einen (oder mehrere) Socks-Server abgewickelt. Ihr eigenes System ist für Außenstehende nicht erkennbar, nur der besonders gesicherte Socks-Server.

Es gibt zwei unterschiedliche Arten, Socks zu verwenden. Die erste und häufigste Methode ist die Verwendung von Socks auf Anwendungsebene. Hier muß dann in jeder Anwendung (wie beispielsweise dem WebExplorer/2 oder Netscape/2) ein Socks-Server eingetragen werden.

Zweitens ermöglicht OS/2 Warp den Einsatz von Socks auf TCP/IP-Stack-Ebene. Die Socks-Konfiguration auf Anwendungsebene werden überflüssig (bzw. dürfen nicht mehr verwendet werden) und Socks muß von den Anwendungen selbst auch nicht mehr unterstützt werden.

Um Socks auf TCP/IP-Ebene benutzen zu können, müssen Sie das Feld *Socks aktivieren* mit einem Haken versehen und die Felder *Socks-Benutzer-ID, Socks-Domäne* und *Socks-Namen-Server* ergänzen. Diese und die Werte für die folgenden Seiten, Direct-Leitwege und Socks-Server erhalten Sie, falls nötig, von Ihrem Netzwerkadministrator.



Abb. 9.16: TCP/IP-Konfiguration, Abschnitt Drucken

Abschnitt 9: Drucken, Druckdienste konfigurieren

Wenn Sie einen Drucker über TCP/IP ansteuern wollen, müssen Sie hier die Felder *Ferner Druck-Server*, *Drucker des fernen Druck-Servers* und *Maximale Anzahl von LPD-Anschlüssen* mit Inhalten versehen. Die Werte werden in den Umgebungsvariablen »SET LPR_SERVER« und »SET LPR_PRINTER« durch die »config.sys« gesetzt.

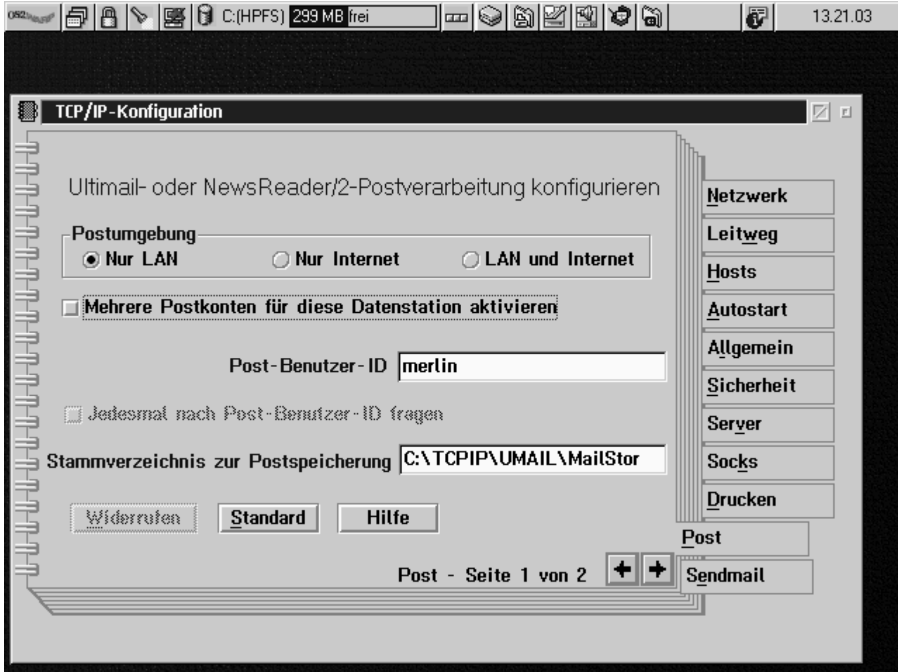


Abb. 9.17: TCP/IP-Konfiguration, Abschnitt Post, Seite 1

Abschnitt 10: Post, Seite 1: Ultimail- oder NewsReader/2-Postverarbeitung konfigurieren

Auf den nächsten Seiten wird die Postumgebung für den NewsReader/2 und das Ultimail/2 konfiguriert. Über die Einstellungen *Nur LAN, Nur Internet* und *lLAN und Internet unter Postumgebung* wird festgelegt, welche Einstellungen für die Verarbeitung von elektronischer Post verwendet werden.

Bei *Nur LAN* werden die postbezogenen Einstellungen aus diesem TCP/IP-Konfigurationsnotizbuch für den NewsReader/2 und Ultimail/2 verwendet.

Mit *Nur Internet* werden die postbezogenen Einstellungen aus den Andere Internet Servicegeber- oder *Netzwerk-Wählfunktion-Konfiguratonsnotizbüchern* verwendet.

Bei *LAN und Internet* werden die Konfigurationsdaten beider Einstellungen verwendet, allerdings werden die Einstellungen aus *Andere Internet* Servicegeber und *Netzwerk-Wählfunktio*n als erstes abgefragt.

Wenn mehrere Personen an Ihrem Computersystem mit UltiMail/2 arbeiten möchten, müssen Sie *Mehrere Postkonten für diese Datenstatio*n aktivieren mit einem Haken versehen. Die *Post-Benutzer-ID* ist entweder die einzige oder die standardmäßig verwendete Post-ID des Benutzers. Tragen Sie hier den Teil vor dem Klammeraffen Ihrer eMail Adresse ein.

Standardmäßig wird hier der Hostname eingesetzt.

*Jedesmal nach der Post-Benutzer-ID frage*n setzen Sie sinnvollerweise, wenn Sie *Mehrere Postkonten* für diese Datenstation aktivieren, um ein versehentliches Verwenden von fremden eMail-Konten zu verhindern. Ultimail/2 fragt dann jedesmal die zu verwendende Post-ID beim Start ab.

Im *Stammverzeichnis zur Postspeicherung* wird die eingehende und ausgehende elektronische Post (zwischen-) gespeichert.

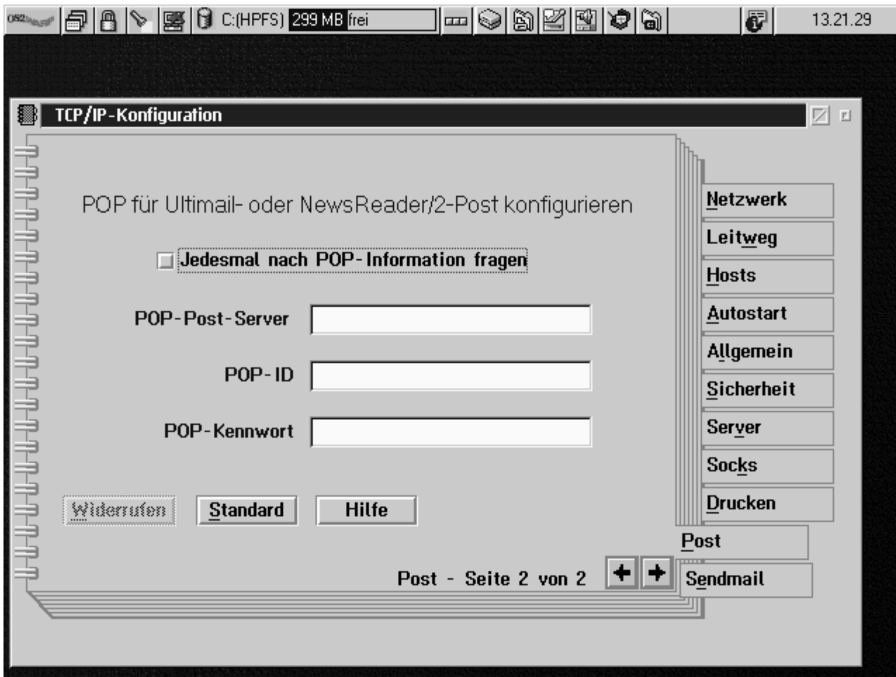


Abb. 9.18: TCP/IP-Konfiguration, Abschnitt Post, Seite 2

Seite 2: POP für Ultimail- oder NewsReader/2-Post konfigurieren

Hier werden die Informationen aufgenommen, die für die Verbindung zum Post-Server notwendig sind.

Wenn Sie auf der Seite 1 das Feld *Mehrere Postkonten für diese Datenstation aktivieren* aktiviert haben, können Sie die POP-Informationen über *Jedesmal nach POP-Information fragen* bei jedem Start von Ultimail/2 abfragen lassen.

POP steht für Post Office Protocol – dieses Protokoll wird von allen Internet-Service-Providern (ISP) akzeptiert und ist für den Empfang, die Zwischenlagerung beim ISP und den Versand von elektronischer Post verantwortlich. Die Konfigurationsdaten erhalten Sie von Ihrem ISP.

Der *POP-Post-Server* ist der Server Ihres ISPs, der Ihre Post empfängt und bis zur Abholung zwischenspeichert.

Die *POP-ID* ist der Benutzername, mit dem Sie bei Ihrem Post-Server registriert sind.

Das *POP-Kennwort* ist das Kennwort für den Post-Server. Diese beiden Werte müssen nicht mit dem Namen und Kennwort, die Sie für die Anmeldung beim ISP verwenden, identisch sein.



Abb. 9.19: TCP/IP-Konfiguration, Abschnitt Sendmail, Seite 1

Abschnitt 11, Seite 1: Sendmail-Parameter konfigurieren

Auf den nächsten zwei Seiten werden die notwendigsten Konfigurationsdaten für das Dienstprogramm »Sendmail« abgefragt und die Dateien »sendmail.cf« und/oder »sendmail.uml« aktualisiert.

Mit *Bei Verwendung von Ultimail/Postverarbeitung von NR/2* werden die eingegebenen Daten in der \mptn\etc\sendmail.uml gespeichert.

Mit *Bei Verwendung einer anderen Postsoftware* werden die eingegebenen Daten in der \mptn\etc\sendmail.cf gespeichert und stehen somit auch anderen Post-Anwendungen zur Verfügung.

Die vorgegebenen Werte in den Feldern *Mqueue-Verzeichnis, Postverzeichnis* und *Postzustellprogramm* können übernommen werden.

In das Feld *Antwortdomäne* ist der Teil **nach** dem Klammeraffen Ihrer Email Adresse einzutragen, zum Beispiel »teamos2.de«. Aus diesem Feld und dem Feld *Post-Benutzer-ID* im Abschnitt *Post.*

Der *SMTP-Post-Gateway* ist das Computersystem, das für das eigentliche Verschicken Ihrer elektronischen Post zuständig ist. Im Normalfall sind dieses Computersystem und der POP-Post-Server identisch.

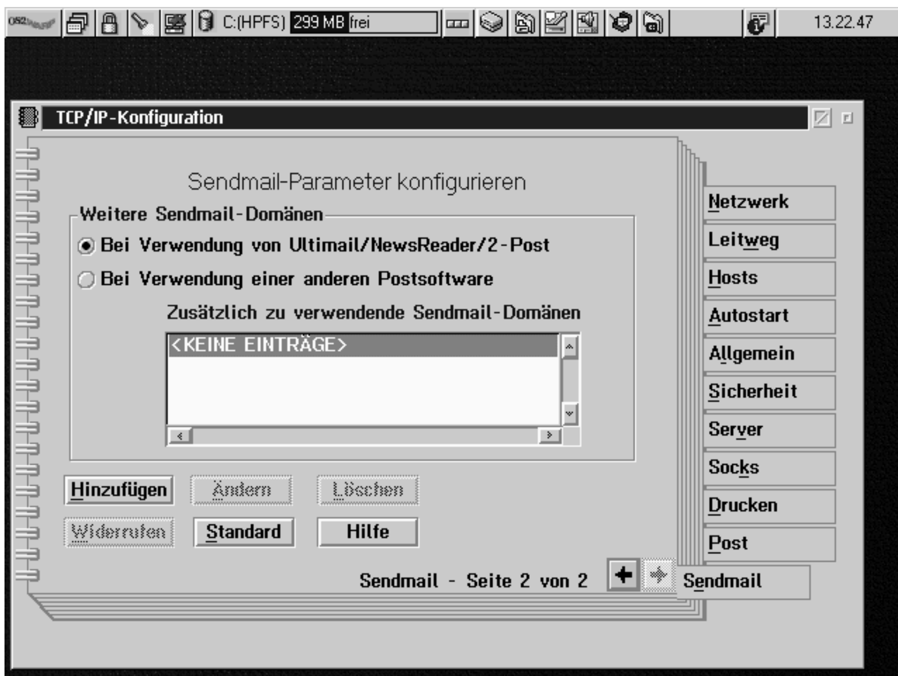


Abb. 9.20: TCP/IP-Konfiguration, Abschnitt Sendmail, Seite 2

Seite 2: Sendmail-Parameter konfigurieren

Wenn Ihr Computersystem sozusagen in mehreren Domänen »zu Hause« ist und in Ihrem LAN Email-Adressen mit unterschiedlichen Antwortdomänen verwendet werden, kann es notwendig sein, über *Zusätzlich zu verwendende Sendmail-Domänen* weitere lokale Domänen einzustellen. Im Normalfall müssen hier allerdings keine Änderungen vorgenommen werden.

9.6.1 Die Arbeitsdateien der TCP/IP-Konfiguration

Vom TCP/IP-Konfigurationsnotizbuch werden die nachfolgenden Dateien verändert bzw. erzeugt:

\config.sys

Hier werden die Netzwerkkartentreiber und Protokolltreiber geladen, sowie die TCP/IP- und MPTS-Umgebung initialisiert. Nachfolgend eine Auswahl der TCP/IP-bezogenen Einträge:

```
SET TZ=mez-1        Setzen der Zeitzone
SET HOSTNAME=merlin Setzen des Hostnamen
SET NLSPATH=C:\MPTN\MSG\NLS\%N;c:\tcpip\msg\dede850\%N;
                    Setzen des Pfads für die verwendete Sprache
SET ETC=C:\MPTN\ETC Setzen des ETC Verzeichnisses
DEVICE=C:\MPTN\PROTOCOL\SOCKETS.SYS Laden des Socket-Treibers
DEVICE=C:\MPTN\PROTOCOL\AFINET.SYS  Laden des IP-Adressen-Treibers
DEVICE=C:\MPTN\PROTOCOL\IFNDIS.SYS  Laden des IP-NDIS-Treibers
RUN=C:\MPTN\BIN\CNTRL.EXE           INET Semaphoren erstellen
CALL=C:\OS2\CMD.EXE /Q /C C:\MPTN\BIN\MPTSTART.CMD >NUL
 Initialisieren der MPTN und TCP/IP Umgebung, Aufruf von \MPTN\BIN\NBSETUP.CMD
oder \MPTN\BIN\SETUP.CMD
DEVICE=c:\tcpip\bin\vdostcp.vdd
                                    Laden des virtuellen IP-Treibers für DOS
DEVICE=c:\tcpip\bin\vdostcp.sys     Laden des IP-Treibers für DOS
RUN=c:\tcpip\bin\VDOSCTL.EXE        DOS INET Semaphoren erstellen
```

\mptn\bin\setup.cmd

Hier wird die TCP/IP-Umgebung initialisiert. Es werden die Netzwerkschnittstellen konfiguriert und die Leitwege gesetzt.

```
route -fh      Alle Leitwege werden gelöscht
arp -f         Alle ARP Informationen werden gelöscht
ifconfig lan0 192.168.0.1 netmask 255.255.255.0
               Die Netzwerkschnittstelle lan0 wird
               eingerichtet, Adresse und Netzwerkmaske gesetzt
route add default 192.168.0.1 1   Der Standardleitweg wird gesetzt
route add net 192.168.0 192.168.0.1 1 netmask 255.255.255.0
               Der Leitweg für das lokale Netz wird gesetzt
ipgate off     IP-Weiterleitung wird nicht aktiviert
```

\mptn\etc\resolv

Hier werden die Domain- und Nameserver-Informationen für die Modemverbindung gesetzt

\mptn\etc\resolv2

Hier werden die Domain- und Nameserver-Informationen für die LAN-Verbindung gesetzt

\mptn\etc\hosts

IP-Namen zu IP-Adressen-Zuordnungen gesetzt

\tcpip\bin\tcpstart.cmd

Starten der TCP/IP-Serverprozesse

\mptn\etc\inetd.lst

Hier werden die über den inetd zu startenden Serverprozesse festgelegt

\mptn\etc\trusers

Benutzerrechte für den ftp-Serverprozeß

\mptn\etc\tcpos2.ini (non-ASCII)

Verschiedene Einstellungen der TCP/IP-Konfiguration

\mptn\etc\socks.env

Hier wird die Socks-Umgebung definiert

\mptn\etc\dhcpd.cfg

DHCP-Client-Informationen

\mptn\etc\rhosts

Berechtigungen für die RSH-Server-Dienste

\mptn\etc\sendmail.cf

Konfigurationsdatei für das Sendmail-Dienstprogramm (allgemein)

\mptn\etc\sendmail.uml

Konfigurationsdatei für das Sendmail-Dienstprogramm (Ultimail/2 und NewsReader/2)

9.6.2 Fehlersuche und Problembehebung bei TCP/IP im LAN

Sie haben alles nach bestem Wissen konfiguriert und trotzdem läuft es nicht so, wie Sie sich es vorgestellt haben? Jetzt ist die Verbindung von unten nach oben, also von Netzwerkkarte bis zur Anwendung zu prüfen.

. Ist die Netzwerkkarte korrekt angeschlossen?

– Wenn Sie mit Koaxialkabel arbeiten: Ist das Netzwerkkabel korrekt terminiert? Können andere Computersysteme auf dem Netz arbeiten?

2. Hat OS/2 einen Anschluß an das Netzwerk?

– Wenn der Netzwerkkartentreiber korrekt geladen worden ist, probieren Sie ein PING auf Ihre eigene IP-Adresse mit »PING <IP-Adresse>«, um zu prüfen, ob TCP/IP korrekt installiert ist. Sollte dies korrekt funktionieren, testen Sie, ob OS/2 auf ein fernes System zugreifen kann mit »PING <IP-Adresse>«. Ist dies nicht möglich, überprüfen Sie Ihre Teilnetzwerkmasken, die IP-Adresse, die Leitwege und Router-Einstellungen.

3. Ist die Applikation korrekt konfiguriert?

– Müssen Sie unter Umständen einen Proxy-Server konfigurieren, um mit dem WebExplorer zu arbeiten? In den Server-Einstellungen des WebExplorers muß ein Proxy mit »http://proxy.domain.de:<portadresse>/« eingetragen werden. Leicht vergißt man hier »http://« und »/« am Anfang und Ende des Eintrages.

9.7 Praxis II: Konfiguration TCP/IP über Modem

von Florian Piekert und Hendrik Fulda

Wenn Sie TCP/IP »nur« über eine Telefonleitung betreiben wollen, haben Sie noch einmal die Wahl: Sie können den »Doppelklick«-Weg ins Internet über IBM gehen oder sich mit etwas mehr Arbeit an jeden anderen Internet-Service-Provider (ISP) anschließen.

Der Weg über IBM hat seine Vorteile: Die Konfiguration besteht im wesentlichen aus der Eingabe der Kreditkartennummer und Adresse. Mit Fragen nach TCP/IP-Adressen, Netzwerkmasken und anderem werden Sie nicht belästigt – alles konfiguriert sich »automagisch«. Außerdem können Sie sich in über hundert Ländern lokal einwählen!

Auf der anderen Seite steht der Preisvergleich mit anderen internationalen oder nationalen ISPs. Auch benötigen Sie für die Nutzung des IBM-Internet-Service eine Kreditkarte. Wir haben im Anhang Auswahlkriterien zur Internetprovider-Wahl zusammengestellt, die Sie für eine Wahl zu Rate ziehen können.

Die Konfiguration des IBM-Internet-Service bedarf nur eines Doppelklicks auf das Objekt *IBM Internet Wählfunktion* im Ordner *Internet Modem*.

9.7.1 Internet ohne Kreditkarte: Andere Internet-Servicegeber

Wenn Sie sich für die Nutzung eines anderen Internet-Service-Providers oder Online-Dienstes entschieden haben, stehen ein paar Konfigurationsschritte an, die allerdings in ihrer Einfachheit gleiches suchen.

Für die Konfiguration eines Internetzuganges benötigen Sie **mindestens** folgende Daten von Ihrem Internet Service-Provider oder Online-Dienst:

– Anmelde-ID (Login Name)

– Kennwort (Login Passwort). Achtung! Meist wird Groß- und Kleinschreibung unterschieden!

– Rufnummer des ISP

– Verbindungsart (Protokoll): SLIP oder PPP

– Bei der Verwendung von SLIP: Eigene IP-Adresse, Ziel-IP-Adresse (Router / Gateway), Netzwerkmaske

– MTU-Größe (meistens 1500)

– Wird van-Jacobsen-Header-Kompression verwendet (meistens ja)?

– IP-Adresse (numerisch!) des Domänennamen-Servers (Domain Name Service (DNS))

– Host-Name (IP-Name) des eigenen Computersystems (nicht zwingend notwendig)

– Domänenname (Domain Name) des eigenen Computersystems

– IP-Name des Nachrichten-Servers (News oder NNTP Server)

Wenn Sie auch Email verwenden wollen:

– IP-Name des Post-Gateway (SMTP)

– IP-Name des POP-Post-Servers (POP3 Server)

– POP-Anmelde-ID (Login Name) und POP-Kennwort für den Post-Server

– Ihre Email-Adresse

9.7.2 IP über das Modem: SLIP und PPP

Da TCP/IP ein Netzwerkprotokoll ist, das für den Einsatz in LAN Netzwerken entwickelt wurde, mußte man für die Übertragung eines TCP/IP-Datenstromes über serielle Verbindungen spezielle Protokolle entwickeln. Mit Hilfe dieser Protokolle können serielle Schnittstellen als ganz normale IP-Netzwerkanschlüsse verwendet werden. Eines der beiden zu diesem Zweck in OS/2 Warp enthaltenen Protokolle wird von jedem ISP unterstützt. Die beiden Protokolle heißen:

SLIP steht für Serial Line IP und ist ein quasi-Standard. SLIP ist das ältere der beiden Protokolle und insgesamt schwieriger zu konfigurieren, da mehr Parameter angegeben werden müssen. Auch muß SLIP eigentlich mit einer festen IP-Adresse betrieben werden, die die meisten ISPs nicht mehr vergeben.

PPP steht für »*Point to Point Protocol*« und ist ein Internet-Standard. PPP ist, was IP-Adressen, Netzwerkmasken und ähnliches angeht, selbstkonfigurierend. PPP ist eigentlich »unterhalb« des zu übertragenden Protokolls angesiedelt. D.h., über PPP können im Prinzip auch andere Protokolle als IP (wie z.B. IPX) übertragen werden. Die meisten PPP-Implementationen unterstützen jedoch im Moment nur TCP/IP.

9.7.3 In der Praxis: SLIP oder PPP?

Im Normalfall wird Ihr Provider PPP als Protokoll verwenden. Wenn Sie vor die Wahl gestellt werden, ist auch PPP zu nehmen, da es einfacher zu konfigurieren ist. Rein technisch ist SLIP bei Verwendung aller Optimierungsmechanismen um ein Winziges schneller, dieser hauchdünne Vorteil ist aber in der Praxis nicht von Bedeutung.

9.7.4 Die TCP/IP-Dienste (Modem) konfigurieren

Um einen Wählzugang zu konfigurieren öffnen Sie den Ordner *Programme* auf der Arbeitsoberfläche, dann den Ordner *Internet (Modem)*, dann *Internet-Dienstprogramme (Modem)* und dann das Objekt *Andere Internet-Servicegeber*. Alternativ können Sie auch *Netzwerk-Wählfunktion* im *Internet (LAN)*-Ordner verwenden.

Hier sind jetzt vier Seiten mit Konfigurationsdaten zu füllen, wobei einige Felder nur bei Bedarf ausgefüllt werden müssen.

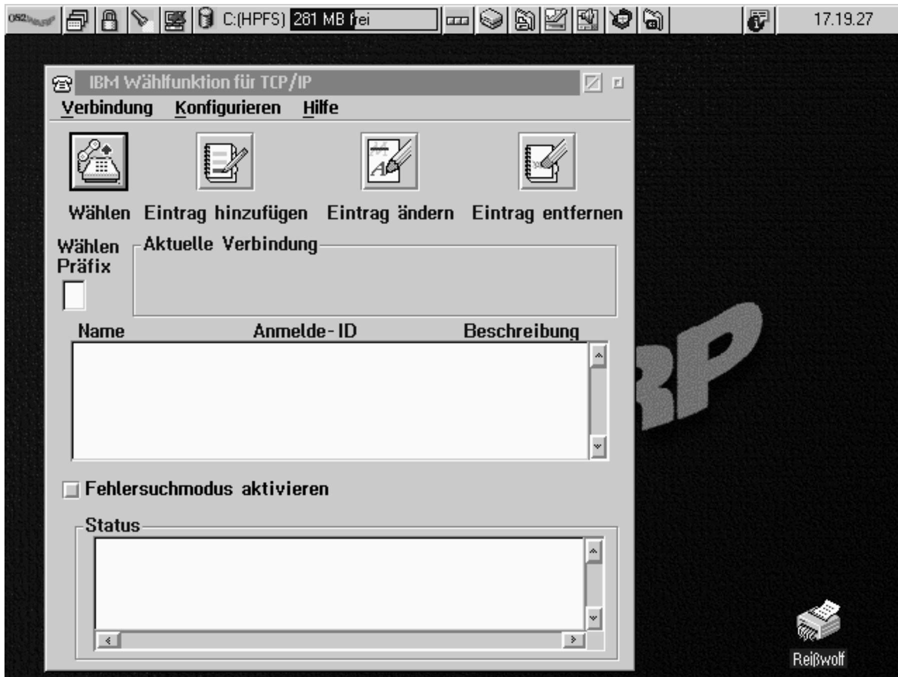


Abb. 9.21: Andere Internet-Servicegeber / IBM-Wählfunktion für TCP/IP

Über *Eintrag hinzufügen* nehmen Sie einen neuen Eintrag auf. Es erscheint ein Konfigurationsnotizbuch, in dem Sie dann die jeweilige Verbindung konfigurieren können. Die Felder, die mit einem Sternchen markiert sind, müssen ausgefüllt werden, die anderen nur bei Bedarf.

Abschnitt 1: Seite 1: Anmeldung

Unter *Name* tragen Sie einen kurzen Namen für diese Verbindung ein – in *Beschreibung* können Sie weitere Informationen unterbringen.

Anmelde-ID

Die Anmelde-ID haben Sie von Ihrem Internet-Service-Provider erhalten. Sie wird auch als Login-Name oder Login-ID bezeichnet. Bei der Anmelde-ID wird im Normalfall zwischen Groß- und Kleinschreibung unterschieden, achten Sie also auf die Schreibweise!

Kennwort

Das zu der Anmelde-ID gehörige Kennwort haben Sie auch von Ihrem ISP erhalten. Auch hier wird zwischen Groß- und Kleinschreibung unterschieden! Das Kennwort wird in der \mptn\etc\tcpos2.ini gespeichert.

Abb. 9.22: Einträge hinzufügen, Abschnitt Anmeldung

Normalerweise ist die Angabe eines Kennwortes bei der Anmeldung notwendig. Die Auswahlbox *Erforderlich* ist also mit einem Haken zu versehen. Ist hier ein Haken und **kein** Kennwort eingetragen, werden Sie bei jedem Verbindungsaufbau nach dem Kennwort gefragt.

Die *Rufnummer* ist die Nummer, die OS/2 Warp wählt, um eine Verbindung herzustellen.

Anmeldeverfahren

Das Feld *Anmeldeverfahren* ist ein Mehrzweckfeld. Es sind verschiedene Eingaben möglich:

NONE als Standardeintrag. OS/2 Warp versucht nicht, eine Anmeldeprozedur durchzuführen. Allerdings wird eine Anmeldeprozedur über das automatische PAP (PPP Authorisation Protocol) Verfahren durchgeführt, wenn der ISP dies unterstützt.

Bei *Kein Eintrag (alle Zeilen löschen)* wird die Standard-Anmeldeprozedur verwendet. OS/2 Warp wartet nun auf die Zeichenfolgen »login:« und »password:« und sendet dann automatische die Inhalte der Felder »Anmelde-ID« und »Kennwort«.

Unter *Dateiname* geben Sie den Namen einer OS/2-Rexx-/Batchdatei oder einer ASCII-Datei an. Eine Rexx-/Batchdatei (*.cmd) wird beim Herstellen der Verbindung ausgeführt und übernimmt die vollständige Anmeldeprozedur. Eine Beispieldatei finden Sie unter \tcpip\bin\annex.cmd. Es können auch Parameter übergeben werden.

Eine ASCII-Datei wird als Anmelde-Antwortdatei interpretiert. Eine Beispiel-Antwortdatei finden Sie unter \tcpip\samples\etc\samples.rsp.

Unter *Ein Anmeldeverfahren* können Sie auch einfach eine Reihe von Zeichenfolgen, auf die OS/2 Warp warten soll, und die Antworten darauf festlegen.

Weiterhin ist die Verbindungsart anzugeben, über die die TCP/IP-Pakete übertragen werden sollen. Welches Verfahren Sie verwenden müssen, erfahren Sie von Ihrem ISP. Wenn Sie die Wahl haben, benutzen Sie PPP, da es leichter zu konfigurieren ist.

Sie können unter *Inaktivitätszeitsperre* auch einen Zeitwert eintragen, nach dessen Verstreichen OS/2 Warp die Verbindung unterbricht, wenn solange keine Daten mehr übertragen wurden.

Abb. 9.23: Einträge hinzufügen, Abschnitt Verbindung

Abschnitt 2: Seite 2: Verbindung

Nur bei der Verwendung von SLIP müssen Sie die Felder *Ihre IP-Adresse*, *Ziel-IP-Adresse* und *Netzwerkmaske* mit den Inhalten, die Sie von Ihrem ISP erhalten haben, ergänzen.

Die MTU-Größe (Maximum Transmission Unit) erfahren Sie auch von Ihrem ISP. Im Normalfall sind die eingetragenen Werte 1006 für SLIP und 1500 für PPP korrekt.

Auch ob VJ-Komprimierung (van Jacobsen Header Compression) zu verwenden ist oder nicht, sagt Ihnen Ihr ISP. Die VJ-Komprimierung von OS/2 Warp erkennt allerdings auch automatisch, ob VJ-Komprimierung möglich ist.

Unter *Domänennamen-Server* ist die **numerische** IP-Adresse des Domain-Name-Service-(DNS) Servers einzutragen, damit Sie später auch symbolische IP-Namen verwenden können.

Ihr Host-Name ist der IP-Name Ihrer Maschine. Dieser Wert muß nicht angegeben werden. Wenn Sie keine statische IP-Adresse haben, ist es unwahrscheinlich, daß für Ihre Maschine ein fester Hostname vergeben wurde.

Unter *Ihr Domänenname* ist die Domain und Toplevel-Domain (z.B. »teamos2.de«) einzutragen, in der Ihr Computersystem sich anmeldet. Sie erfahren auch diese Information von Ihrem ISP.

Abb. 9.24: Einträge hinzufügen, Abschnitt Server

Abschnitt 3: Seite 3: Server

Unter *Standard-Server/-Hosts* sind die IP-Namen oder IP-Adressen der standardmäßig zu verwendenden Server anzugeben. Das Feld *Nachrichten-Server* (News / NNTP) wird vom NewsReader/2, *Gopher-Server* vom Gopher-Client und *WWW-Server* vom WebExplorer/2 ausgelesen.

Die *Post-Server-Information* werden von Ultimail/2 und vom NewsReader/2 verwendet.

Der *Post-Gateway* wird auch als SMTP-Host oder -Gateway bezeichnet und ist meistens mit dem *POP-Post-Server* identisch. Die *Antwortdomäne* ist der Teil **nach** und die *Antwort-ID (Post)* der Teil **vor** dem Klammeraffen in Ihrer Email-Adresse. Die *POP-Anmelde-ID* ist unter Umständen identisch mit der *Antwort-ID (Post)* und wie das *POP-Kennwort* möglicherweise abhängig von Groß- und Kleinschreibung.

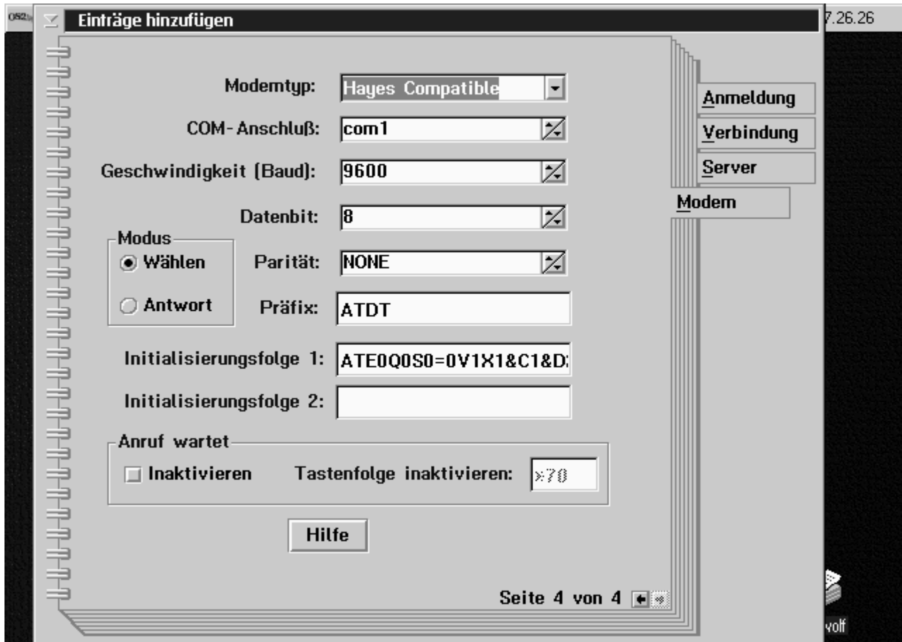


Abb. 9.25: Einträge hinzufügen, Abschnitt Modem

Abschnitt 4: Seite 4: Modem

Hier wird das Modem konfiguriert. *Modemtyp*, *COM-Anschluß* (serielle Schnittstelle) und *Geschwindigkeit* sind entsprechend anzupassen. *Datenbit* (8) und *Parität* (Keine) sind in den Standardeinstellungen korrekt.

Das *Präfix* ist mindestens abhängig von dem Wahlverfahren, das Sie benutzen: *ATDT* für Tonwahl und *ATDP* für Pulswahl. Die *Initialisierungsfolge 1* und *Initialisierungsfolge 2* werden für Ihr Modem entsprechend voreingestellt, bedürfen aber unter Umständen einer Modifikation. Schlagen Sie dazu in Ihrem Modem-Handbuch die entsprechenden Befehle nach.

Unter *Anruf wartet* besteht die Möglichkeit, die Tastenfolge (Wahlfolge) zur Inaktivierung der Anrufwartefunktion Ihrer Telefonanlage zu definieren. Wenn Sie eine Telefonanlage mit dem Leistungsmerkmal *Anklopfen* (oder ähnlich) besitzen, wird das Anklopfsignal die Datenverbindung stören.

Sobald Sie alle notwendigen Parameter eingegeben haben, sollte einer Verbindung zum Internet nichts mehr im Weg stehen.

9.7.5 Fehlersuche und Problembehebung bei SLIP- und PPP-Verbindungen

Sie haben alles nach bestem Wissen konfiguriert und trotzdem läuft es nicht so, wie Sie sich es vorgestellt haben? Jetzt ist die Verbindung von unten nach oben, also von Modemverbindung bis zur Anwendung, zu prüfen.

Versuchen Sie folgende Fragen der Reihe nach zu klären:

1. Ist das Modem richtig angeschlossen?

– Können Sie mit einer anderen Anwendung unter OS/2 Warp (!) auf das Modem zugreifen? Verwenden Sie dazu HyperAccess oder ein anderes Terminalprogramm. Wenn nicht, überprüfen Sie die Einstellungen auf der letzten Seite (Modem).

2. Greift das IBM Internet Wählprogramm richtig auf das Modem zu?

– Wenn Sie die Verbindung herstellen wird in einem Status-Fenster der Wählvorgang angezeigt. Antwortet das Modem auf die Initialisierungsbefehle mit *OK*?. Wenn nicht, überprüfen Sie die Einstellungen auf der letzten Seite des Eintrages.

3. Wird ein Kontakt zwischen Ihrem Modem und dem Modem des ISP hergestellt?

– Die »Carrier detect«-Lampe Ihres Modems brennt nach einem erfolgreichen Verbindungsaufbau. Sollte dies nicht der Fall sein, überprüfen Sie die Telefonnummer (Seite 1) und die Modemeinstellungen (Seite 4).

4. Meldet sich Ihr Computersystem korrekt beim ISP an?

– Im Status-Fenster können Sie den Dialog zwischen der IBM-Wählfunktion und dem ISP verfolgen. Werden die richtigen Antworten gegeben? Wenn nicht, überprüfen Sie Ihr Einwahlscript.

5. Meldet sich Ihr Computersystem wirklich korrekt beim ISP an?

– Wenn Sie im Status-Fenster angezeigt bekommen, daß eine Verbindung aufgebaut worden ist und das Einwahlscript beendet wurde, kann es sich um eine Fehlinterpretation handeln. Bricht die Verbindung nach erfolgreicher Abarbeitung des Einwahlscriptes oder bei Aufruf des ersten Internetprogramms ab, überprüfen Sie Ihr Einwahlscript und vor allem Ihr Passwort. Beim Passwort wird Groß- und Kleinschreibung unterschieden!

6. Nur SLIP: Sind die Routen richtig gesetzt?

– Öffnen Sie eine OS/2-Kommandozeile und geben Sie »PING <IP-Zieladresse>« ein. Bekommen Sie eine Antwort? Wenn nicht, überprüfen Sie die Verbindungsdaten auf Seite 2.

7. Ist der Domain Name Service korrekt konfiguriert?

– Wenn Sie zwar verbunden sind, aber keine anderen Systeme erreichen, prüfen Sie zuerst, ob überhaupt eine Verbindung zum Domain-Name-Service-Server besteht. Öffnen Sie eine OS/2-Kommandozeile und geben Sie »PING <IP-Adresse des DNS-Servers>« ein. Bekommen Sie eine Antwort? Wenn nicht, überprüfen Sie die Verbindungsdaten auf Seite 2. Wenn Sie eine Antwort bekommen, wiederholen Sie die Prozedur und verwenden an Stelle der IP-Adresse den IP-Namen des News-Servers oder eines anderen Systems. Wenn Sie keine Antwort bekommen, überprüfen Sie die Einstellungen auf Seite 2.

9.8 Überblick TCP/IP über ISDN

von Florian Piekert und Hendrik Fulda

TCP/IP kann auch über ISDN genutzt werden. Hierzu gibt es im Prinzip drei Möglichkeiten.

Üblicherweise wird eine ISDN-Karte über einen Treiber namens CAPI (Common ISDN Application Programming Interface) angesprochen. Dieser Treiber wird – wie die NDIS-Treiber der Netzwerkkarten – vom Hersteller geliefert.

Die erste Möglichkeit, TCP/IP über ISDN zu nutzen, ist die Verwendung eines NDIS-Treibers anstelle der CAPI. Die ISDN-Karte stellt sich OS/2 Warp dann wie eine gewöhnliche Netzwerkkarte dar. Dies wird jedoch sehr selten eingesetzt und ist auch meistens mit zusätzlichen Kosten verbunden, da NDIS-Treiber von den ISDN-Karten-Herstellern oft als Zusatz vertrieben werden.

Die zweite Variante verwendet auf der einen Seite die CAPI und auf der anderen die ganz normale IBM-TCP/IP-Wählfunktion (SLIPPM). Da aber die IBM-Wählfunktion nur auf serielle Schnittstellen zugreifen kann, muß ein weitere Treiber auf die CAPI aufgesetzt werden, um eine serielle Schnittstelle und ein Hayes-kompatibles Modem zu emulieren.

Beispiele für solche ISDN-zu-COM-Treiber sind die Programme »Isdncom« der Firma Mikado, »Capicom« (Shareware) und »Cfos/2« (Shareware). Sie stellen eine weitere serielle Schnittstelle zur Verfügung, emulieren gegenüber dem System ein Hayes-kompatibles Modem und stellen zum Teil noch weitere Funktionalitäten wie rudimentären Anrufbeantworterfunktionen etc. zur Verfügung.

Die dritte und (meistens) effektivste Methode ist ein Programm, das direkt auf die CAPI zugreift und dem System eine TCP/IP-Netzwerkschnittstelle (sl0) zur Verfügung stellt, auf die direkt zugegriffen werden kann.

Dieses Programm heißt »Isdn-pm« und liegt als Shareware vor. Die neueste Version ist jeweils auf »ftp://ftp.uni-freiburg.de/« in /pub/pc/os2/isdn/ zu finden.

»Isdn-pm« ist nach Meinung der Autoren die einfachste und schnellste Art, TCP/IP über ISDN zu nutzen. Isdn-pm unterstützt die Protokolle IP over X75, M_LAPB, HDLC_UI, HDLC und CISCO, sowie SLIP, PPP/HDLC und PPP/X.75. PAP oder asynchrone Login-Scripts sind möglich.

9.9 Die Kommandozeilenprogramme zur TCP/IP-Konfiguration

von Florian Piekert und Hendrik Fulda

Wie bereits unter »Die Arbeitsdateien der TCP/IP-Konfiguration« nachzulesen, arbeitet OS/2 Warp mit verschiedenen Kommandozeilenprogrammen, um die eigentliche TCP/IP Umgebung herzustellen. Diese wollen wir hier kurz erläutern.

\mptn\bin\arp.exe

ARP steht für »Adress Resolution Protocol« und stellt die Zuordnung zwischen IP-Adressen und Netzwerkkartenadressen im lokalen LAN her.

Dafür wird eine dynamische Tabelle angelegt, die sogenannte ARP-Table. Hier speichert OS/2 Warp IP-Adressen und die zugehörigen Adressen der Netzwerkkarten.

Im normalen Betrieb sind die ARP-Tabellen und das ARP-Dienstprogramm nicht weiter von Interesse. Nur wenn Sie zum Beispiel feststellen möchten, ob OS/2 Warp auch korrekt auf das LAN zugreifen kann oder wenn es notwendig ist, daß Sie die ARP-Tabellen direkt manipulieren müssen, brauchen Sie das ARP-Dienstprogramm.

Rufen Sie »arp.exe« ohne Parameter auf, um eine Übersicht über die Parameter zu bekommen. Rufen Sie »ARP -a« auf, um eine Liste der IP-Adressen und zugehörigen Netzwerkkartenadressen auf Ihrem LAN zu erhalten.

Wenn Sie »ARP -a« direkt nach dem Neustart aufrufen, ist die Tabelle leer. Erst wenn Sie beispielsweise mit PING ein Zielsystem angesprochen haben, wird dessen Netzwerkkartenadresse über das ARP Protokoll in Erfahrung gebracht und in die Tabelle eingetragen.

\mptn\bin\ifconfig.exe

IFCONFIG steht für »Interface Configuration«. Mit dem Ifconfig-Dienstprogramm werden unter OS/2 Warp die Netzwerkschnittstellen eingerichtet und konfiguriert.

Rufen Sie »Ifconfig.exe« ohne Parameter auf, um eine Übersicht über die Parameter zu bekommen. Rufen Sie »Ifconfig lan0« auf, um die Konfiguration Ihrer ersten Netzwerkschnittstelle angezeigt zu bekommen. Hier kann man zum Beispiel die Rundsendeadresse (Broadcast) auslesen.

Sie können Ihre Netzwerkschnittstellen dynamisch ein- und ausschalten (UP | DOWN), sowie Parameter verändern.

\mptn\bin\route.exe

Das Dienstprogramm »Route« wird zum Erstellen, Ändern und Löschen von Leitwegen (Routen) verwendet. Die Einstellungen können im laufenden Betrieb verändert werden.

Rufen Sie »route.exe« ohne Parameter auf, um eine Übersicht über die Parameter zu bekommen. Um die bestehenden Routen anzuzeigen, verwenden Sie das Dienstprogramm »Netstat« mit dem Parameter »-R«.

Mit »ROUTE ADD DEFAULT 192.168.0.1 1« setzen Sie beispielsweise den Standardleitweg (default route) auf den Router mit der IP-Adresse 192.168.0.1.

\mptn\bin\netstat.exe

Mit »Netstat« können verschiedene Statusanzeigen, aber auch TCP/IP-Statistiken angezeigt werden. Nestat ist ein wichtiges Dienstprogramm, um Informationen über bestehende Verbindungen (NETSTAT -S), gesetzte Leitwege (Routen, NETSTAT -R) und verwendete Adressen (NETSTAT -A) auszulesen.

Rufen Sie »netstat.exe« ohne Parameter auf, um eine Übersicht über die Parameter zu bekommen. Rufen Sie »netstat -R« auf, um alle gesetzten Routen zu erfahren.

Benutzen Sie Netstat, um die Einstellungen der Netzwerkschnittstellen und Leitwege (Routen) zu überprüfen.

9.9.1 TCP/IP unter der DOS- und Windows-Emulation

Bei OS/2 Warp ist ebenfalls ein TCP/IP-Stack für die DOS- und Windows-Emulationen vorhanden und wird bei Installation des TCP/IP-Protokolles automatisch installiert.

Um unter der DOS- und Windows-Emulation auf die TCP/IP-Dienste von OS/2 Warp zugreifen zu können, müssen beim OS/2-Systemstart virtuelle TCP/IP-Treiber geladen werden, da das unter OS/2 vorhandene TCP/IP-Protokoll von den virtuellen DOS- und Windows-Sitzungen benutzt werden soll.

Dazu werden in der \»config.sys« drei Programme bzw. Treiber geladen. Es handelt sich dabei um die Dateien »vdostcp.vdd«, »vdostcp.sys« und »vdosctl.exe«, die sich alle in \tcpip\bin befinden.

Windows-Anwendungen greifen in der Regel auf eine Winsock.dll zu. Unter OS/2 Warp ist diese für die Windows-Emulation unter \tcpip\dos\bin zu finden. Achten Sie darauf, daß diese Datei nicht von Windows-Installationsprogrammen für TCP/IP-Anwendungen überschrieben wird.

Zuletzt ist wichtig, daß in der Datei »\autoexec.bat« die ETC-Umgebungsvariable mit »SET ETC=\TCPIP\DOS\ETC« korrekt gesetzt ist und in diesem Verzeichnis die Dateien »hosts«, »protocol«, »resolv« und »services« korrekt vorliegen. Sie müssen mit den Dateien im \mptn\etc-Verzeichnis übereinstimmen.

Mit den DOS- und Windows-Dienstprogrammen \tcpip\dos\bin\ping.exe und \tcpip\dos\bin\wping.exe können Sie die Funktionsfähigkeit der DOS- und Windows-TCP/IP-Emulation überprüfen.

9.10 Beschreibung der mit Warp V4 ausgelieferten TCP/IP-Programme

von Florian Piekert und Hendrik Fulda

9.10.1 Weltweit elektronisch Post versenden

Es gibt mittlerweile für OS/2 einige gute Dienstprogramme, die es dem Benutzer ermöglichen, auf den Internetdienst Electronic Mail> (Email) zuzugreifen. Dabei werden standardmäßig mit Warp V4 zwei Dienstprogramme ausgeliefert, drei weitere befinden sich zusätzlich auf der Bonus- und Lotus-CD. Das eine Programm, das fest zum normalen Lieferumfang von OS/2 gehört, ist »UltiMedia Mail/2 Light«. Das andere ist der »IBM WebExplorer«. Die auf den zusätzlichen CDs vorhandenen Programme stammen zum einen von Innoval System Solutions, Inc. (dabei handelt es sich um den »PostRoadMailer« in der Green- und Blue-Edition) aus den USA und von Lotus (Lotus Mail). Die beiden Programme von Innoval sind auch auf Sharewarebasis zu bekommen. Mit dem einen Programm (Green-Edition) kann der Benutzer seine Email über das POP3- oder SMTP-Protokoll verarbeiten, mit dem anderen kann er auf das sogenannte PROFS zugreifen, wenn man einen entsprechenden Zugang sein eigen nennt. Die Programme von Innoval und Lotus werden hier nicht weiter erläutert, da diese nicht zum eigentlichen Bestandteil von Warp V4 gehören.

Um es vorweg zu nehmen: UltiMedia Mail/2 Light ist nicht der Weisheit letzter Sch(l)uß. Es gibt drei empfehlenswerte hervoragende Email-Clients für die direkte Benutzung im Internet für OS/2. Zum einen ist das der PostRoad-Mailer von Innoval, der auf der Bonus-CD mitgeliefert wird, zum anderen handelt es sich um MR/2 ICE (MailReader/2 Internet Cruiser Edition) und das mittlerweile sehr komfortabel und unverzichtbar gewordene PMMail, das seit kurzem in der Version 1.53 vorliegt. Zu beziehen sind diese Programme unter anderem per ftp von »ftp.teamos2.de« aus dem Verzeichnis pub/OS2/Hobbes/Network/TCPIP/Mail/Clients. Dort befindet sich auch der PostRoadMailer in einer Demoversion. Diese Programme sind zur Zeit allerdings nur in englischer Sprache vorhanden, wobei dieses für UltiMedia Mail/2 Light und NotesMail ebenfalls zutrifft und daher nicht das Hindernis darstellen sollte.

Weiterführende Information und Softwareupdates zu diesen Programmen können unter den WWW-URLs »www.southsoft.com/pmmail.html, www.innoval.com« oder »nick.secant.com/mr2ice.html« gefunden werden.

UltiMedia Mail/2 Light

Bei »UltiMedia Mail/2 Light« handelt es sich um einen Email-Client, der in der Lage ist, sowohl Mail von der lokalen Festplatte oder per POP3-Protokoll vom einem Mailserver eines Internet-Service-Providers zu verarbeiten. Damit Email auf dem lokalen Computersystem (auf der lokalen Platte) ankommen kann (über SMTP aus dem Internet oder dem lokalen Netz eingeliefert), muß auf dem OS/2-Computersystem ein Daemonprozess gestartet werden, der auf Port 25 auf Verbindungen wartet. Unter Warp V4 erledigt diese Aufgabe der OS/2-Port von Sendmail. Sendmail wird hier nicht weiter erklärt, dazu gibt es ein Buch, daß sich nur über die kryptische Konfiguration von Sendmail ausläßt.

Das Versenden von neu geschriebenen Nachrichten erfolgt ebenfalls über Sendmail per SMTP (simple mail transfer protocol). Allerdings ist es standardmäßig so konfiguriert, daß die erzeugten neuen Nachrichten direkt an das Mail-EXchanger-Computersystem des Empfängers ausgeliefert werden und nicht etwa der Mailserver des Providers dazu benutzt wird. Es gibt allerdings bei Sendmail die Möglichkeit, einen Smart- oder Relayhost zu konfigurieren. Im Usenet gab es vor langer Zeit dazu eine Diskussion, in der einige Personen beschrieben haben, wie Sendmail zu konfigurieren wäre, damit dieses funktioniert. Das Ergebnis war, daß es bei einigen Benutzern funktioniert hat, bei der Mehrheit aber nicht.

Dieses hat zum Ergebis, daß das Ausliefern der Nachrichten wesentlich länger dauert, als wenn es nur zum lokalen Mailserver des Internetproviders gehen würde. Sollte es nicht möglich sein, die Nachricht direkt auszuliefern (aufgrund von eventuell auftretenden Timeouts zum Beispiel), so wird dieses später erneut versucht, im schlimmsten Fall wird die Nachricht gar nicht versandt.

Um UltiMedia Mail/2 Light zu benutzen, werden im Ordner *Programme* die Ordner *Internet (Modem)* oder *Internet (LAN)* je nach Internetanschluß ausgewählt. In diesen Ordnern befindet sich entweder eine Referenz oder der Ordner *UltiMedia Mail/2 'Lite'*. Wird dieser Ordner geöffnet, so befinden sich dort fünf Objekte. Wird eines der vier Objekte außer dem Objekt *Information* geöffnet, so wird als erstes eine Abfrage nach den Benutzerdaten getätigt, sofern diese nocht nicht vollständig über das Objekt *TCP/IP Konfiguration* konfiguriert wurden.

Abb. 9.26:
Benutzerdaten eingeben

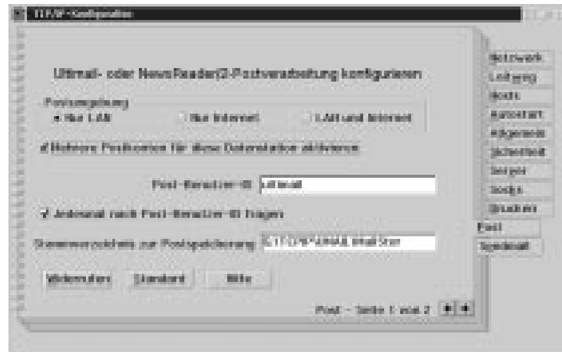


Abb. 9.27: Benutzerdaten über das Objekt TCP/IP-Konfiguration definieren

Wenn der Internetzugang über das lokale Netzwerk (LAN) erfolgt, so sollte nur der Punkt *nur LAN* für den Mailempfang auf der Seite *Post* ausgewählt werden, obwohl auch Internet-Email empfangen werden kann. Wird *LAN und Internet* oder *Internet* ausgewählt, so wird bei jedem Öffnen eines UltiMedia Mail/2 Light-Objektes das Programm *LinkUp* gestartet, das ausgibt, daß keine Modemverbindung ins Internet besteht (was ja auch logisch ist, weil der Zugang über das Netzwerk stattfindet). Durch Klicken auf *Nicht verbinden* kann dieses jedoch übergangen und UltiMedia Mail/2 Light dennoch gestartet werden. OS/2 Warp versteht unter dem Begriff *Internet* hier eine Modemverbindung in das Internet und nicht die Verbindung über das Netzwerk.

Mit Hilfe des TCP/IP-Konfigurationsobjektes werden die Angaben über den POP3-Post-Server (POP3-Mailserver, POP3 steht für Post Office Protocol 3, ein Protokoll, um auf einem Server liegende elektronische Post mit einem entsprechenden Dienstprogramm abzuholen) des Providers, die POP-ID (den POP3-Loginnamen) und das POP3-Kennwort (Password) für diesen POP3-Mail-Zugang gemacht. Diese Daten erhält der Benutzer in der Regel von seinem Internet-Service-Provider (ISP).

Wird bei *UltiMedia Mail/2 Light* das Paßwort eingegeben, wenn es nicht über das TCP/IP-Konfigurationsobjekt konfiguriert wird, so muß das eingegebene Paßwort bestätigt werden.



Abb. 9.28: Paßwortbestätigung

Wird UltiMedia Mail/2 Light zum Beispiel über das Objekt *Mail Cabinet* gestartet, so

erscheint eine Übersicht über die vorhandenen Email-Ordner. Beim ersten Starten wird nur der *SentMail*-Ordner (gesendete Nachrichten) angezeigt. Über das Menü *Windows* können die bis jetzt vorhanden Ordner aufgerufen werden.

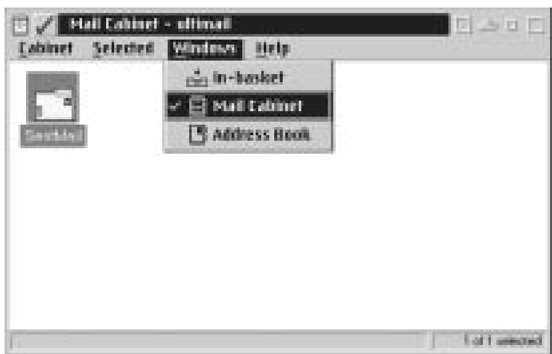


Abb. 9.29: MailCabinet

Um die auf dem Mailserver des Providers liegenden Nachrichten zu laden oder um neue Nachrichten lesen zu können, wird entweder das Objekt *In Basket* geöffnet oder über den Menüpunkt *Windows In Basket* der Posteingangsordner geöffnet. UltiMedia Mail/2 Light startet automatisch eine Verbindung zum Mailserver des Providers (sofern eine aktive Internetverbindung vorliegt) und lädt die neuen Nachrichten, die auf dem Mailserver liegen, herunter und speichert sie auf der Festplatte. Diese neu angekommenen Nachrichten werden jetzt im *In Basket* aufgelistet.

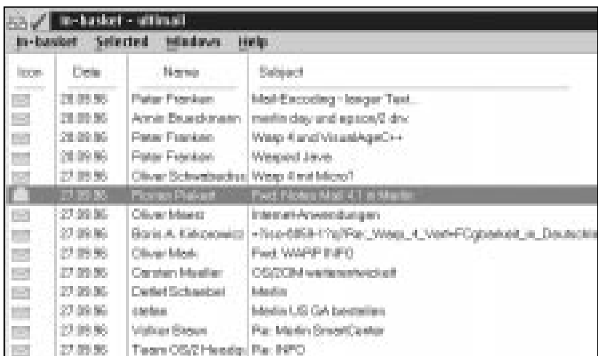


Abb. 9.30:
Der In-Basket-Ordner

Neue Nachrichten können nun gelesen werden, indem die durch Doppelklick geöffnet werden oder durch die Auswahl von *Read New Mail* aus dem *In Basket*-Menü. Wird letzteres benutzt, so werden alle noch nicht gelesenen neuen Nachrichten geöffnet und der Benutzer ist in der Lage, über die Pfeiltasten in der Titelleiste des Nachrichtenbrowsers zwischen den Nachrichten zu wechseln.

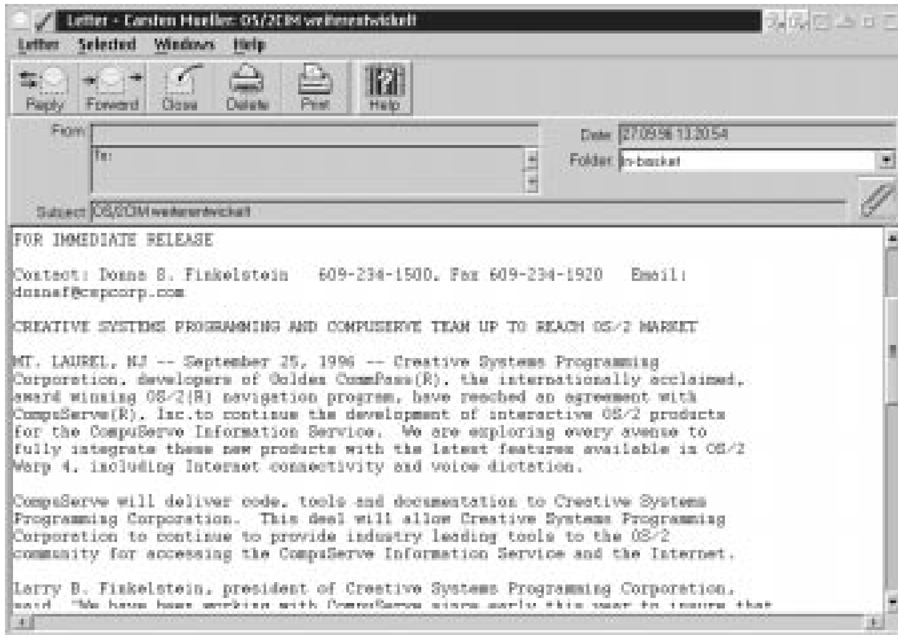


Abb. 9.31: UltiMedia Mail/2 Light-Nachrichtenbrowser

Im Nachrichtenbrowser ist ersichtlich, ob diese Nachricht *Attachments* (angehängte Dateien) besitzt. Dieses wird durch das Klicken auf das Heftklammersymbol erreicht. Dort ist auch zu erkennen, aus wie vielen Attachments (Anhängsel) oder Teilen diese Nachricht besteht. Wird dort mehr als nur ein Teil angezeigt, hat diese Nachricht Attachments. Der gewünschte Teil kann durch Doppelklick aktiviert werden. Bei Bilddateien wird das Bild angezeigt, bei Binärdateien kann ein Programm gewählt werden, das diese Datei verarbeiten kann. Wenn es sich um eine ausführbare Datei handelt, so kann diese ausgeführt werden.

Um jetzt auf eine Nachricht zu antworten, wird einfach der Button *Reply* ausgewählt. Der Inhalt der gelesenen Nachricht, auf die sich die Antwort bezieht, wird leider nicht automatisch zitiert.

Neue Nachrichten können entweder über das Objekt *New Letter* erzeugt werden, oder indem der Untermenüpunkt *New Mail* ausgewählt wird.

Als erstes sollte ein Empfänger dieser Nachricht angegeben werden, damit diese auch später den gewünschten Empfänger erreicht.



Abb. 9.32: Mail mit Attachments

Zusätzlich kann jeder Nachricht noch ein Betreff gegeben werden (Subject), damit es möglich ist, zum Beispiel schneller den Inhalt der Nachricht zu finden, wenn das Subject eine Zusammenfassung mit einem Schlagwort darstellt.

Um in diese neue Nachricht neben dem normalen Briefinhalt in Textform Bild-, Audio- oder sonstige Binärdateien zu integrieren, werden einfach die über dem Textfeld stehenden entsprechenden Buttons benutzt und die zu integrierende Datei ausgewählt. Es kann dann noch spezifiziert werden, was mit den entsprechenden *Attachments* geschehen soll, wenn der Empfänger diese Nachricht öffnet.

Handelt es sich bei dem gewünschte Attachment um eine Bilddatei, so kann man diese mit UltiMedia Mail/2 Light auch noch bearbeiten, wobei diese Bildbearbeitungsfunktionen natürlich nicht sehr umfangreich sind.

Während des Testens mit Bild-, Audio- und sonstigen Binärdateien ist mir UltiMedia Mail/2 Light des öfteren abgestürzt, was für mich aber keineswegs verwunderlich war. UltiMedia Mail/2 Light ist in den Kreisen der internetnutzenden OS/2-Anwender nicht gerade ein beliebtes oder vielverwendetes Programm. Auf manchen Systemen soll es hingegen gerüchteweise sogar fehlerfrei funktionieren.

Das Programm ist als solches nicht sehr komfortabel. Es unterstützt diverse internationale Standards nicht vollständig oder nicht korrekt (als bestes Beispiel sei hier die MIME-Unterstützung genannt). Bei der Testnachricht mit einem attachten Bild, die ich mir selber mit UltiMedia Mail/2 Light zugesandt habe, war UltiMedia Mail/2 Light hinterher nicht in der Lage, diese Nachricht auch vernünftig darzustellen oder auszuwerten. Die Grafik wurde nicht korrekt dekodiert und stattdessen kam nur ein in Text kodiertes Etwas dabei heraus.

Um UltiMedia Mail/2 Light sauber vom OS/2-Warp-V4-System zu entfernen, sollten folgende Schritte eingehalten werden: Zuerst sollten aus der »config.sys« die Einträge des Verzeichnisses, in dem sich UltiMedia Mail/2 Light befindet (normalerweise befinden sich die dazugehörigen Dateien im tcpip\umail-Verzeichnis), aus den Angaben LIPPATH, PATH und HELP entfernt werden. Danach sollte ein Reboot erfolgen, um sicherzustellen, daß die vorhandenen UltiMedia-Mail/2-Light-DLLs nicht mehr vom Betriebssystem benutzt werden. Anschließend sollte das tcpip\umail-Verzeichnis komplett und die Datei »umail.hlp« aus dem tcpip\help-Verzeichnis gelöscht werden. Jetzt können alle UltiMedia-Mail/2-Light-Objekte auf der Arbeitsoberfläche entfernt werden (den ganzen Ordner *UltiMedia Mail/2 Lite* einfach auf den Reißwolf ziehen und notfalls mit *Ja* bestätigen. Wenn sich einige Dateien nach dem Reboot immer noch nicht löschen lassen, kann dieses durch einen weiteren Reboot gelöst werden. Um auch die Einträge aus den OS/2-Ini-Dateien zu entfernen, kann ein Sharewaretool wie »Uni-Maint« verwendet werden.

Email versenden mit dem IBM-WebExplorer

Mit dem IBM-WebExplorer ist es auch im begrenzten Maßstab möglich, elektronisch Post zu versenden. Dieses beschränkt sich auf reine ASCII-Texte, da diese per Hand eingegeben werden müßen. Das attachen von Binärdateien ist nicht möglich. Ebenfalls ist es nicht möglich, mit dem IBM WebExplorer Email zu lesen.

Um einem anderen Teilnehmer eine Email zu senden wird der Menüpunkt *Navigieren Post senden* ausgewählt. Dort werden dann die Angaben zu dem Empfänger und dem Thema (Subject) definiert. Anschließend kann in dem dafür vorgesehenen Fenster die Eingabe des Textes erfolgen.

Ist dieses abgeschlossen, so wird versucht, diese Nachricht zu versenden. Wie auch bei UltiMedia Mail/2 Light erfolgt die Zustellung der Email direkt an das MX-Computersystem des Zielpartners und nicht über ein eventuell vorhandenes Mail-Relay des Providers. Anders als bei UltiMedia Mail/2 Light besteht hier nicht die Möglichkeit diese Nachricht zu speichern, falls der Versandversuch nicht erfolgreich verlaufen ist. Um es erneut zu probieren muß das Formular erneut abgesendet werden. Wird das Programm beendet, so ist auch diese Nachricht gelöscht.

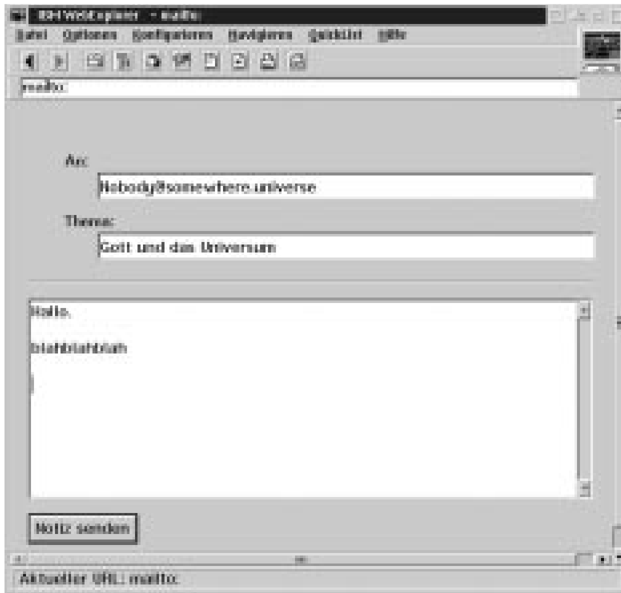


Abb. 9.33:
IBM WebExplorer

9.10.2 Die schwarzen Bretter im Internet

Mit Warp V4 werden zur Zeit zwei Dienstprogramme mitgeliefert, mit denen das Benutzen der Newsgroups im Usenet/Internet ermöglicht wird. Zunächst ist das einmal der »NewsReader/2«, ein PM-Programm. Der andere mitgelieferte Client ist der »IBM WebExplorer«, mit dem ebenfalls auf Newsgroups zugegriffen werden kann. Sobald der Netscape Navigator/2 fertig gestellt ist, wird dieser ebenfalls mit OS/2 Warp ausgeliefert. Auch dieses Programm ermöglicht den Zugriff auf die Newsgroups.

Die Funktionalität unterscheidet sich ein wenig bei beiden Dienstprogrammen. Generell ist der NewsReader/2 besser geeignet, auf die Newsgroups zuzugreifen als der WebExplorer oder als Netscape, da letztere eigentlich als WorldWideWeb-Clients entwickelt wurden und die Newsgroupfunktionalität später hinzuaddiert wurde um einen Allround-Client für Informationssuche im Internet zu haben.

NewsReader/2

Der mit OS/2 Warp mitgelieferte Client ist in der Lage, mit einem Newsserver eine Verbindung aufzubauen. Der Newsserver erwartet Anfragen auf Port TCP 119 (NNTP). Beim ersten Starten des NewsReader/2 muß ein Newsserver über *Aktionen-Nachrichten-Server ändern* selektiert werden.

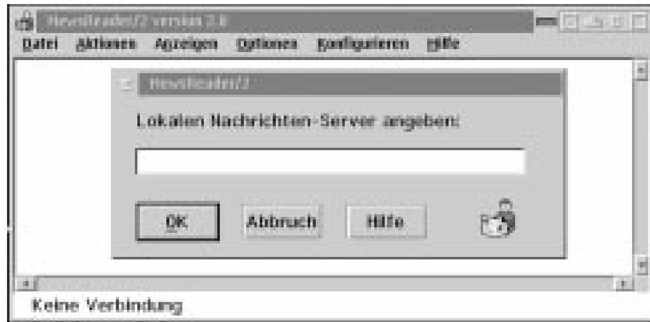


Abb. 9.34: Welchen Newsserver hätten Sie denn gerne?

Um nach Informationen in den Angeboten der Newsgroups zu suchen, muß als erstes eine Liste der auf diesem gewählten Newsserver vorhandenen Newsgroups vom Newsserver geholt werden. Bei langsamen Verbindungen mit einem Modem kann dieses durchaus ein wenig länger dauern, abhängig vom Angebotsspektrum des Newsservers.

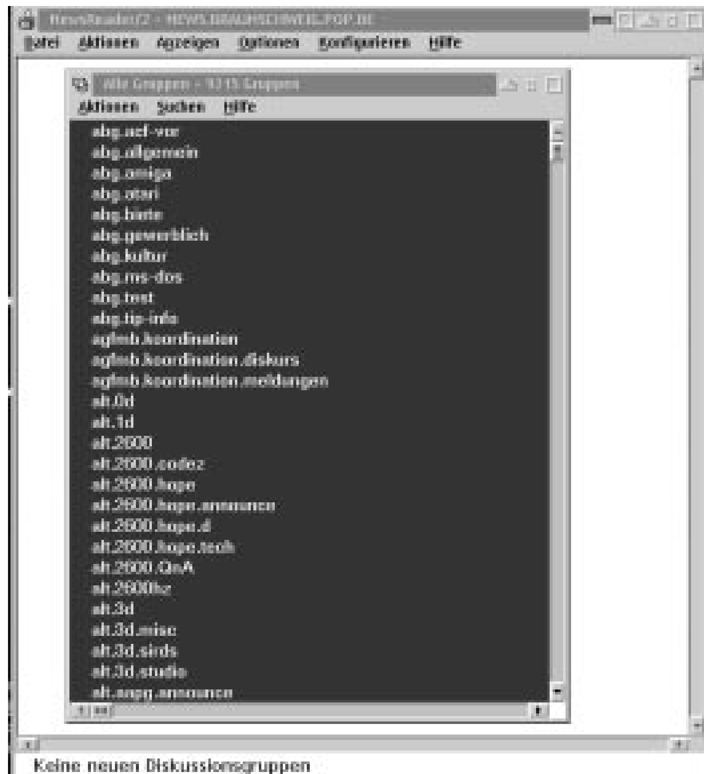


Abb. 9.35: Die Newsgruppenliste

Über den Menüpunkt *Anzeigen Alle Gruppen – XXXX* kann nun die geladene Übersicht der auf diesem Newsserver vorhandenen Newgruppen angezeigt werden. Die Zahl XXXX gibt die Anzahl der verschiedenen Newsgruppen an.

Aus dieser Liste können nun gewünschte Newsgruppen ausgewählt werden. Woher soll der Benutzer nun aber wissen, um was für Diskussionen es in dieser ausgewählten Gruppe geht? Im Normalfall spiegeln die Namen der Newsgruppen den Inhalt wieder. Zum Beispiel gibt es in der Newsgruppe »comp.os.os2.announce« Ankündigungen zu OS/2-Software von Herstellern, Usergrouptreffen oder einfach nur Neuigkeiten rund um OS/2. Das Bestellen dieser Newsgroups ist recht einfach. Es wird der Menüpunkt *Aktionen-Gruppe(n) hinzufügen* ausgewählt. Dadurch werden jetzt die Gruppen den eventuell bereits vorhandenen bestellten Newsgruppen hinzugefügt und vom Newsserver die Anzahl der Artikel in den entsprechenden Newsgroups geladen.

Jetzt, wo eine Auswahl an gewünschten Newsgroups vorhanden ist, sollte die weitere Konfiguration des NewsReader/2 erfolgen. Damit sollte gewährleistet werden, daß Artikel auch korrekt unter dem Namen und der Email-Adresse des Autors erstellt und in das Internet eingespeist werden.

Über die Menüpunkte *Optionen* und *Konfigurieren* ist die Konfiguration möglich. Über den Punkt *Konfigurieren-Beiträge versenden* werden die Angaben zum Autoren bzw. zum Benutzer getätigt.

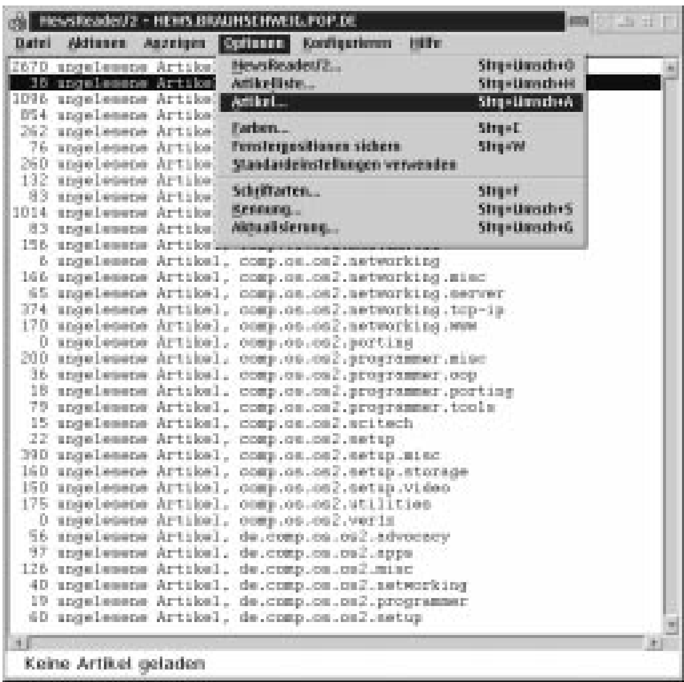


Abb. 9.36: Konfigurationsmöglichkeiten

Die Konfiguration der einzelnen Optionen ist sicherlich Geschmackssache und wird daher hier nicht näher erläutert. Die Online-Hilfe gibt recht aufschlüssig Rat über die einzelnen Optionen.

Der NewsReader/2 ist selbständig in der Lage, nach einem bestimmten Zeitintervall auf neu auf dem Newsserver eingeganene Newsartikel zu prüfen. Ist dieses der Fall, so wird die Liste der bestellten Newsgruppen automatisch aktualisiert und angepaßt.

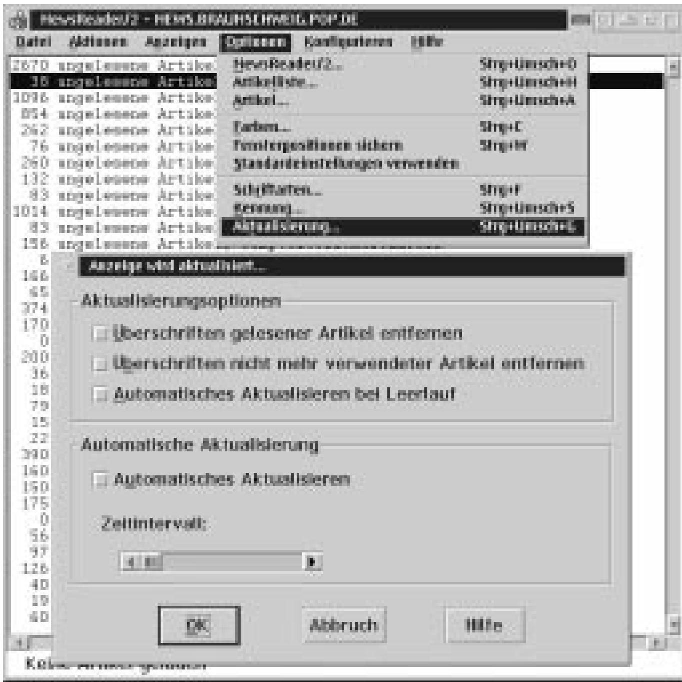


Abb. 9.37: Möglicher automatischer Refresh

Sind nun alle Konfigurationen erfolgt (es sollte nur einmal erforderlich sein, alle diese Konfigurationen zu machen), so kann das eigentliche Lesen der Newsgroups und somit die Informationssuche losgehen. Dazu wird als erstes eine Newsgruppe ausgesucht, die für den Benutzer Interessant erscheint. Durch Doppelklicken auf den Namen der Gruppe wird die Artikelliste dieser Newsgruppe geladen. Dieses kann je nach Artikelaufkommen in dieser Gruppe länger oder weniger lange dauern.

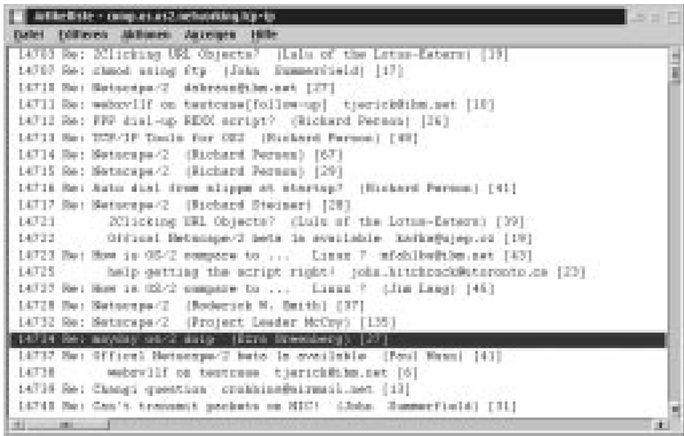


Abb. 9.38: Artikelliste vom comp.os.os2.networking.tcp-ip

Je nach Konfiguration des NewsReader/2 kann diese Liste jedoch anders aufgebaut sein. Um nun einen Artikel aus dieser Newsgroup zu lesen, wird dieser markiert und mit [Enter] oder Doppelklick geladen.



Abb. 9.39: Newsartikel

Das Lesen weiterer Artikel kann einfach durch Drücken von [Enter] erfolgen, wenn dieses entsprechend vorher konfiguriert wurde.

Um auf Artikel zu antworten und selber in diese Newsgroup zu posten, wird der Menüpunkt *Aktionen-Beitrag versenden* ausgewählt. Soll dem Autor des Artikels jedoch eine private Nachricht als Email gesendet werden, so wird statt *Antwort versenden* der Punkt *Antwort verschicken* ausgewählt, wobei allerdings der Wortlaut seitens der IBM verwirrend gewählt ist.

Ist das Antwortschreiben fertig gestellt, so kann dieses über den Menüpunkt *Beitrag versenden* losgesendet und somit in das Internet eingespeist werden. Wurde bei diesem Artikel jedoch ein Fehler gemacht oder möchte der Autor diesen Artikel lieber doch nicht weiterverbreiten, so

besteht die Möglichkeit, diesen Artikel zu löschen (dabei spricht man im Internet/Usenet von *Canceln*). Dieses geschieht über eine Kontrollnachricht, die über das Menü erstellt werden kann. Dazu wird die erstellte Nachricht zum Lesen ausgewählt und dann über das Menü *Aktionen* oder die Taste [F9] der Punkt *Beitrag löschen* ausgewählt. Wird die Bestätigung mit *Ja* beantwortet, so wird der geschriebene Artikel auf allen Newsservern, auf denen er verteilt wurde, wieder gelöscht und nicht weiter verteilt.



Abb. 9.40: Das Löschen von eigenen Nachrichten

Mit dem NewsReader/2 ist der Benutzer leider nicht in der Lage, mehrere Newsserver zu verwalten. Über Handarbeit mit Umherkopieren der relevanten INI- und Newsgruppen-Files ist dieses zwar schon lösbar, gestaltet sich aber nicht gerade komfortabel.

Um mit dem NewsReader/2 Newsartikel lesen zu können, muß eine Online-Verbindung in das Internet zum Newsserver bestehen. Sobald ein Artikel zum Lesen selektiert wird, baut der NewsReader/2 eine Verbindung zum Newsserver auf und lädt diesen Artikel herunter. Selbst wenn der gleiche Artikel mehrfach hintereinander gelesen werden soll, wird er immer wieder aufs neue vom Newsserver geladen.

Es ist somit nicht möglich, mit dem NewsReader/2 offline News zu lesen, um etwa Telefonkosten zu sparen. Dafür gibt es aber Sharewareprogramme, die diese Funktionalität erfüllen und mit denen noch ein wenig komfortabler als der NewsReader/2 gearbeitet werden kann. Eine solche Beispielsoftware befindet sich in der NeoLogic-NetSuite-Internetprogrammsammlung, das per ftp unter anderem vom ftp-Server des TeamOS/2 Deutschland »ftp.TeamOS2.de« im Verzeichnis »pub/OS2/Hobbes/Network/TCPIP/NeoLogicNetSuite/« als »nns45e.zip« zu beziehen ist.

Der NewsReader/2 hat in seiner eingedeutschten Version noch einige grobe Fehlübersetzungen. Zum Beispiel wird in der Artikelübersicht unter dem Menüpunkt *Anzeigen Sofort aktualisieren* fälschlicherweise die Auswahl zwischen *Aktueller Artikel* und *Alle Artikel* gemacht, was

natürlich nicht ganz korrekt ist. Richtig wäre gewesen, wenn die Auswahl zwischen *aktueller Gruppe* und *alle Gruppen* zu treffen wäre. Bleibt zu hoffen, daß diese kleinen Schönheitsfehler in einem der nächsten FixPacs korrigiert werden.

Newsgroups lesen mit dem IBM-WebExplorer

Mit dem IBM-WebExplorer ist es ebenfalls möglich, auf die Informationsangebote der Newsgroups im Internet zuzugreifen. Es ist allerdings nicht ganz so komfortabel wie mit dem NewsReader/2. Um den Zugriff auf einen Newsserver zu starten, muß dieser unter *Configure-Servers* an der Stelle *Newsserver* eingetragen sein. Dann wird über den Menüpunkt *Navigieren* die Option *Diskussionsgruppenmanager* ausgewählt.

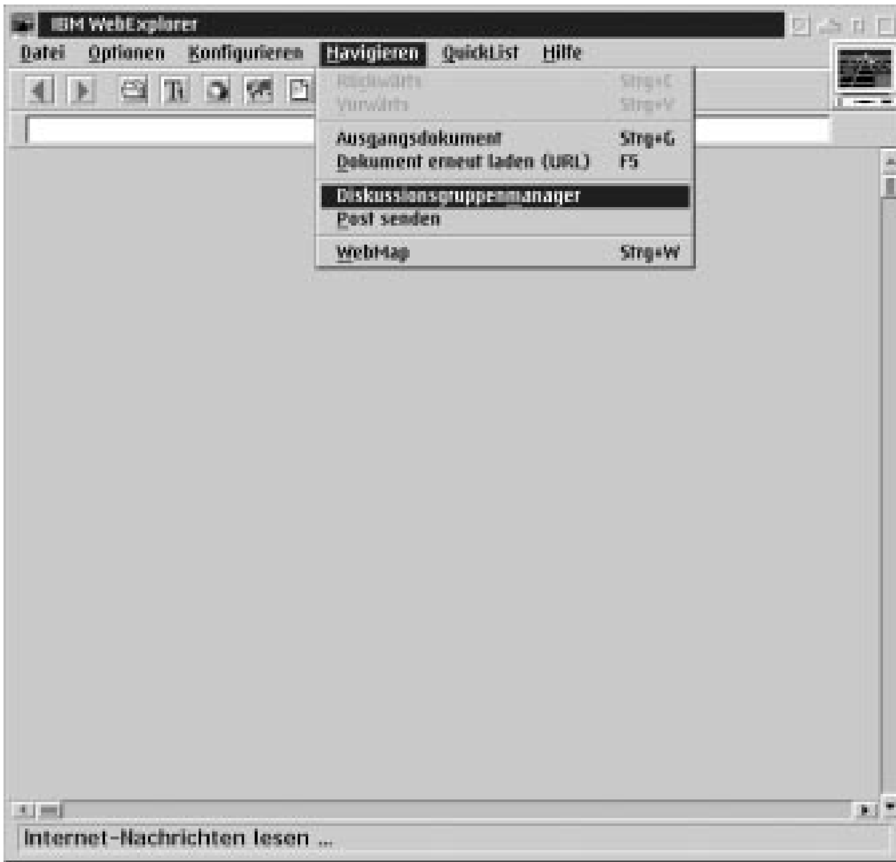


Abb. 9.41: Lesen von Newsgroups mit dem IBM-WebExplorer

Der IBM-WebExplorer ist in der Lage, das vorhandene Newsgroupfile zu benutzen, das der NewsReader/2 anlegt, um die selektierten Newsgruppen zu kennzeichnen. Ist keines vorhanden, so besteht die Möglichkeit, mit dem WebExplorer Newsgruppen zu selektieren und auch die Liste aller vorhandenen Newsgruppen zu laden.

Um einen Artikel aus einer dieser gewählten Newsgruppen zu lesen, wird zuerst die Newsgruppe durch einen Klick ausgewählt. Bei vielen gewünschten Gruppen ist dieses recht langwierig, weil es nicht möglich ist, mehr als eine Gruppe gleichzeitig zu selektieren.

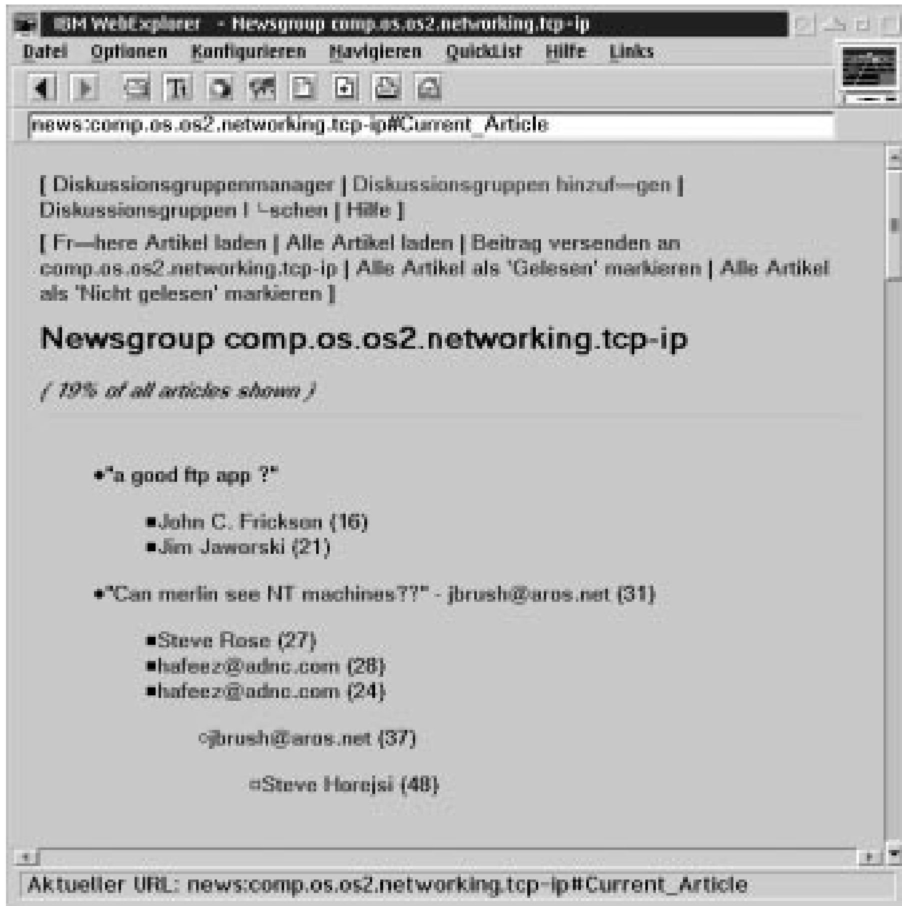


Abb. 9.42: Artikelübersicht

In der Artikelübersicht wird jetzt der gewünschte Artikel selektiert und geladen.

Die Artikel werden gethreaded angezeigt. Das bedeutet, daß Artikel, die zueinander in Bezug stehen, in der Artikelübersicht untereinander versetzt angezeigt werden. Dadurch ist es leicht

zu erkennen, wenn zu einem Artikel ein Kommentar oder eine Antwort existiert. Die Verkettung erfolgt durch die Verknüpfung der Message-IDs und Referenzen darauf der einzelnen Artikel. Würde die Verkettung über die Titel (Subjects) der Artikel laufen, so ist es recht einfach, andere Artikel gleichen Titels fälschlicherweise in diesen Thread einzuordnen.



Abb. 9.43: Gethreadete Artikel

Um selber Antworten oder eigene Artikel in die entsprechend gewählte Newsgruppe zu senden wird einfach der Punkt *Beitrag versenden an Diskussionsgruppe(n)* oder *Antwort versenden* angewählt, wobei mit dem Punkt *Antwort versenden* eine Follow-Up-Nachricht in diese Diskussionsgruppe erzeugt wird. Leider wird der alte Artikel nicht zitiert, was sogenannte Follow-Ups (Nachfolgeartikel) erschwert. Der NewsReader/2 ist in der Lage, den alten Artikel zu zitieren.

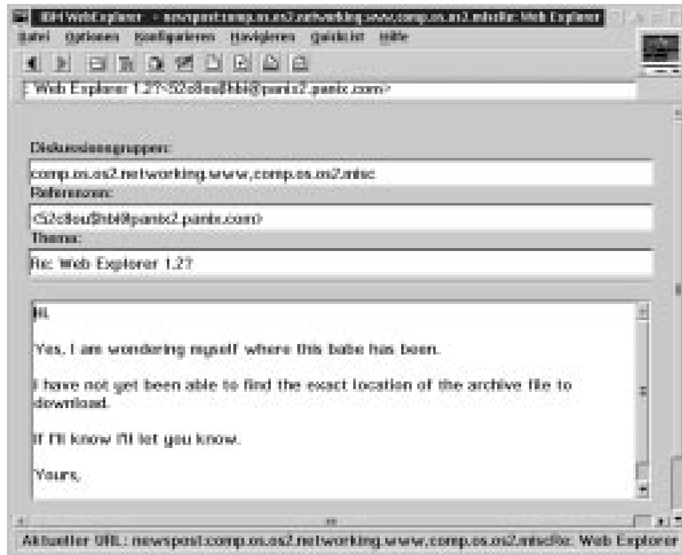


Abb. 9.44: Eigene Artikel posten

Es ist mit dem IBM-WebExplorer leider nicht möglich, gesendete Artikel wieder zu löschen oder zu canceln. Die einzige Möglichkeit, dieses dennoch zu tun, ist es, per Hand die Nachricht entsprechend zu schreiben, damit diese einer Cancel-Nachricht gleicht.

Mit dem WebExplorer ist der Benutzer in der Lage, schnell Artikel in Newsgroups lesen, ohne sich großartig Gedanken über die Konfiguration eines Newsclients zu machen. Aber zum effizienten Nutzen der Newsgroups ist dieses Programm nicht ausgelegt.

Newsgroups lesen mit NetscapeNews

Es gibt beim Netscape Navigator die Möglichkeit, ein neues Browser-Fenster zu öffnen, das als Newsreader fungiert. Beim ersten Starten dieses Clients ist noch kein Newsserver konfiguriert. Dieses geschieht über den Menüpunkt *File | Open News Host*. Es wird einfach die Adresse des gewünschten Newsservers angegeben und NetscapeNews baut eine auf dem NNTP-Protokoll basierende Verbindung zu diesem Newsserver auf.

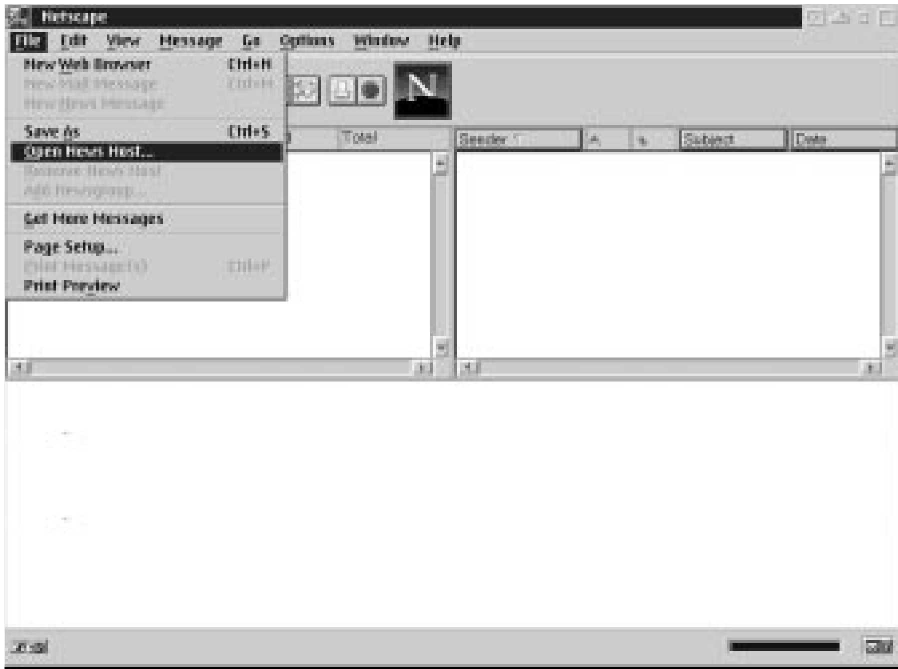


Abb. 9.45: NetscapeNews Newsserverauswahl

NetscapeNews hat bereits beim ersten Start drei Gruppen selektiert. Diese bestehen aus den Newsgruppen »news.announce.newusers«, »news.newuser.questions« sowie der Gruppe »news.answers«. Diese Newsgruppen dienen der Information über das Verhalten in den Newsgroups. Sie sind hauptsächlich an neue Benutzer der Newsgroups im Internet gerichtet. Dort findet man Informationen wie das ungeschriebene Verhaltensgesetz, die Netiquette des Internets. Diese Gruppen können natürlich deselektiert werden. Die eigentlich gewünschten Newsgruppen können dann selektiert werden.

Die Übersicht über die vorhandenen Newsartikel und Threads ist recht übersichtlich gestaltet, vorrausgesetzt das NetscapeNews-Fenster ist groß genug. Die Artikel sind hier auch wieder gethreaded dargestellt, wobei die Verknüpfung ebenfalls über die Message-IDs erfolgt.

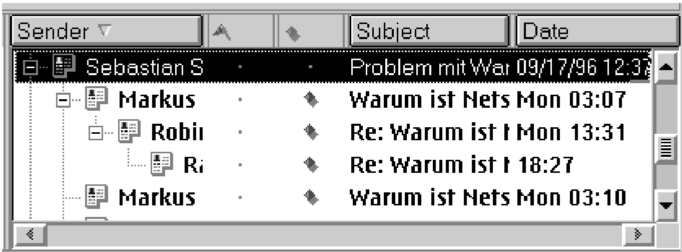


Abb. 9.46: NetscapeNews Artikelübersicht

Wird auf einen Artikel in der Übersicht geklickt, so erscheit dieser im unteren Bereich des NetscapeNews-Fensters und kann gelesen werden.

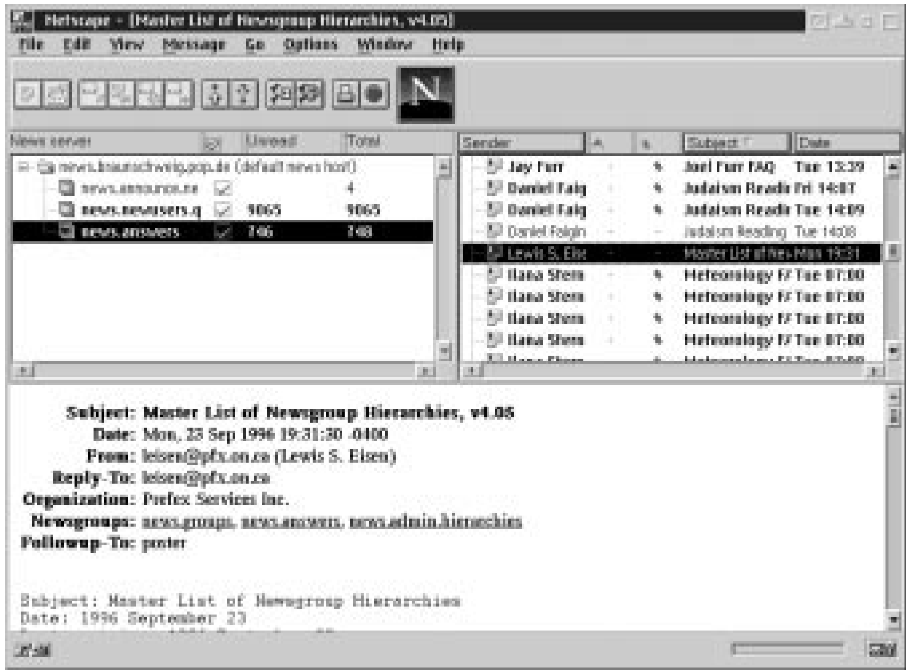


Abb. 9.47: NetscapeNews

Auch mit NetscapeNews ist es nicht möglich, Artikel für eine offline-Verarbeitung zu markieren. Zwar können viele Artikel selektiert und somit geladen werden, aber wenn ein neuer Artikel selektiert wird, so wird er erneut vom Newsserver transferiert.

9.10.3 Das WWW unter Warp V4 benutzen

Für Warp V4 gibt es jetzt zwei verschiedene OS/2-Dienstprogramme (auch Browser genannt) für das WorldWideWeb. Mit dem System ausgeliefert wird immer noch der altbekannte IBM-WebExplorer. Der neue Client ist der kürzlich erst auf OS/2 portierte NetScape Navigator in der Version 2.02. Dieser wird zwar nicht mitgeliefert, der IBM-WebExplorer besitzt aber einen Link, mit dem es möglich ist, sofort nach Erhalt des Systems den Netscape Navigator auf die Festplatte zu laden.

Warum zwei verschiedene Browser?

Im Internet sind sehr viele WWW-Seiten mit Erweiterungen versehen, die nicht im HTML 3.0 Standard spezifiziert sind (zum Beispiel Frames, Java Applets, Java Script). Der IBM-WebExplorer baut auf diesem HTML-3.0-Standard auf und bringt teilweise auch eigene Erweiterungen (zum Beispiel animierte Statusanzeige) mit, teilweise ist er auch in der Lage, Erweiterungen von anderen Browsern darzustellen (Tabellen zum Beispiel). Die anderen vorhandenen Erweiterungen werden einfach nicht dargestellt, was bei einigen Informationsangeboten zu nichts aussagenden Informationsseiten führt.



Abb. 9.48: Frames

Der Netscape Navigator hingegen hat eine große Palette eigener Erweiterungen etabliert, die der WebExplorer teilweise leider nicht darstellen kann (Frames als das berüchtigte Beispiel). Mit dem Erscheinen einer OS/2-Version des Netscape Navigators haben sich diese und viele andere Problematiken für alle OS/2-Benutzer mit einem Schlag verflüchtigt.

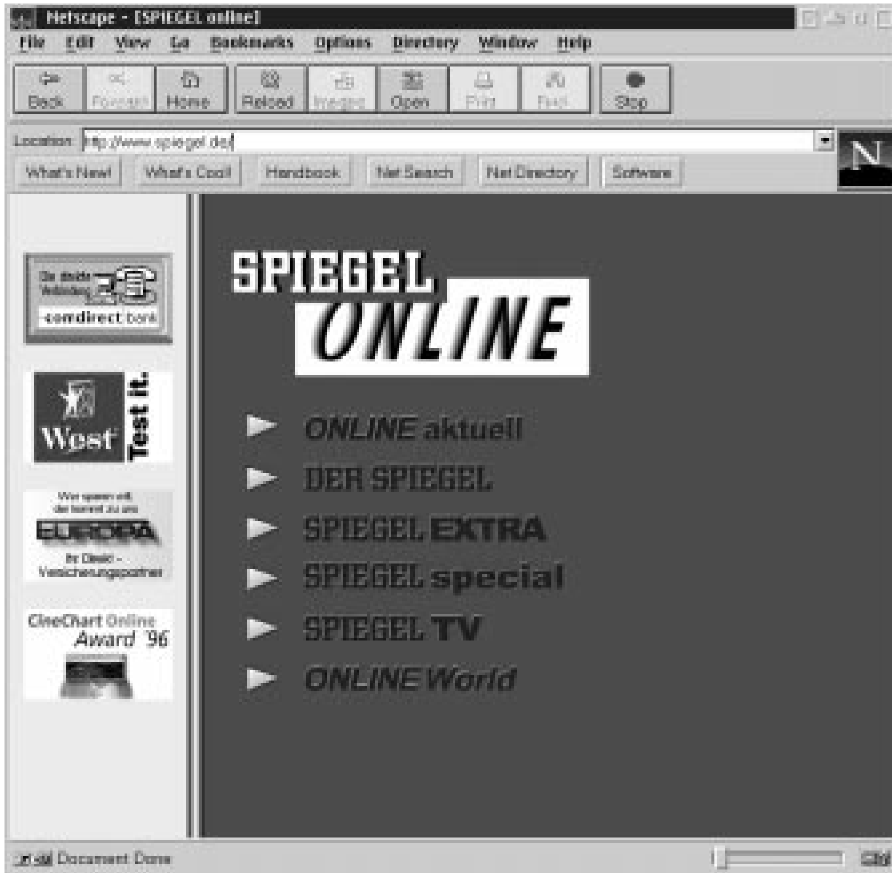


Abb. 9.49: Frames, so und nicht anders ...

Die derzeit frei verfügbare Betaversion Release Beta 1A des Netscape Navigators ist bereits recht stabil und kann im Alltagsgebrauch durchaus verwendet werden. Abstürze sind das eine oder andere Mal zwar noch vorhanden (meistens auch im Zusammenhang mit Java). Diese Betaversion verfügt allerdings noch nicht über sämtliche Funktionalitäten der Windows-Counterparts.

Leider ist der Netscape Navigator zur Zeit auch noch nicht in der Lage, über Sprache navigiert zu werden, URL-Objekte per Drag&Drop zu erstellen oder zu diesen zu springen, wenn ein

solches Objekt auf den Netscape Navigator fallen gelassen wird. All dieses sollte jedoch mit dem Erscheinen des endgültigen Releases der Fall sein.

NetscapEmail ist zur Zeit ebenfalls noch nicht benutzbar. Mit NetScapeNews können zwar schon Newsgroups gebrowsed und Artikel gelesen werden, aber es ist noch nicht möglich, eigene Artikel zu posten oder Artikel weiterzumailen.

Zusammenfassend gesagt: Es ist nicht leicht, etwas über dieses Produkt zu schreiben, da sehr viele Funktionen zum jetzigen Zeitpunkt einfach noch nicht integriert sind und somit Aussagen nicht getroffen werden können. Die Browserfunktionalität, was integrierte HTML-Befehle angeht, zu beschreiben, ist sicherlich ebenfalls nicht Thema dieses Kapitels.

Den IBM-WebExplorer zu beschreiben ist sicherlich auch keine sinnvolle Maßnahme, da sich die IBM mit der Entwicklung des Netscape Navigators gerüchteweise verpflichtet hat, die Entwicklung des WebExplorers einzustellen. Warum also ein Produkt beschreiben, das früher oder später völlig von der Bildfläche verschwinden wird? Netscape ist einfach in der Lage, mehr Erweiterungen anzuzeigen und zählt im Internet als der WWW-Browser-Standard, noch meilenweit vor dem Konkurrenzprodukt aus Redmond. Die Funktionen die der WebExplorer im Zusammenhang mit der Objektorientierung und OS/2 Warp hat, werden hoffentlich in den Netscape Navigator integriert werden, sodaß der WebExplorer wirklich überflüssig wird.

Netscape verfügt zum jetzigen Zeitpunkt schon über die Möglichkeit des Dokument- und Bildercachings, wie der Benutzer es von den Versionen der anderen Plattformen gewohnt ist. Dieser Cache wird im Gegensatz zum Cache des WebExplorers nicht beim Beenden des Programms gelöscht. Vielmehr existiert eine Datenbank, die Protokoll über die einzelnen geladenen Dateien führt und so beim nächsten Start des Netscape Navigators direkt die Seiten aus dem Cache liefern kann, was beim WebExplorer nie der Fall gewesen ist. Selbst mehrere verschiedene Sitzungen des Netscape Navigators greifen auf ein und die selbe Cache-Datenbank zurück. Beim IBM-WebExplorer hat jede Sitzung ihre eigene Cache-Datenbank erstellt. Beim IBM-WebExplorer ist es auch der Fall gewesen, daß jede neue Sitzung gleichbedeutend dem Starten eines neuen WebExplorers war. Beim Netscape Navigator wird lediglich ein neues Fenster geöffnet. Dadurch sollte diese Vorgehensweise ressourcensparender sein.

Es sind noch viele andere Unterschiede zu nennen, doch was letztendlich in der entgültigen Version des Netscape Navigators davon vorhanden ist, kann jetzt noch niemand mit absoluter Sicherheit sagen.

IBM-WebExplorer

Ein paar nützliche Tips zum IBM-WebExplorer gibt es an dieser Stelle dennoch, da dieser WWW-Browser sicherlich eine Zeit lang von den meisten Internet nutzenden OS/2-Benutzern weiter verwendet wird

Parameter für den IBM-WebExplorer

Es besteht durch Parameter die Möglichkeit, den IBM-WebExplorer ein wenig zu konfigurieren. Diese Parameter sind ebenfalls in der Online-Hilfe beschrieben, aber die Mehrheit der Benutzer wird diese sicherlich nicht gelesen haben.

Der Parameter »-q« verhindert das Anzeigen des Logos beim Starten und der Meldung, ob der WebExplorer wirklich beendet werden soll. Diese Abfrage ist in den meisten Fällen ein wenig störend, verhindert jedoch das versehentliche Beenden und somit den Verlust der bis dahin aufgebauten Cache-Datenbank.

Mit dem Parameter »-t 8« wird dem WebExplorer mitgeteilt, daß jener statt der standardmäßig benutzten **vier Threads** mit **acht Threads** arbeiten soll. Dadurch erhöht sich die Verarbeitungsgeschwindigkeit bei Seiten mit vielen Bildern. Es wird auch gemunkelt, daß der WebExplorer so stabiler läuft als nur mit vier Threads.

Durch den Parameter »-p« wird es dem WebExplorer ermöglicht, die Farbpalette zu kontrollieren, sodaß auf Computersystemen mit einer 256-Farben-Darstellung für den WebExplorer immer die geeignetesten Farben bereitstehen, um eine optimale Anzeige der Bilder zu garantieren. Wird diese Angabe nicht gemacht, so wird die zuletzt geladene oder die Systemfarbpalette benutzt.

Soll eine andere Ini-Datei als die sich im etc-Verzeichnis befindliche benutzt werden, so ist dieses über den Parameter »-i« gefolgt von der Lokation der Ini-Datei möglich. So ist es realisierbar, verschiedene Quicklists zu verwalten, ohne das sich die unterschiedlichen WebExplorer-Sitzungen in die Quere kommen.

Als letzter Parameter kann dem WebExplorer eine URL (Uniform Resource Locator) übergeben werden, die dieser beim Programmstart laden soll. Dabei ist es egal, ob es sich um eine ftp-URL oder eine WWW-URL oder um eine html-Datei auf der lokalen Festplatte handelt.

Animationen im WebExplorer-Statusfenster

Es ist möglich, die Status-Animation im WebExplorer zu verändern. Dazu gab es bereits einen Wettbewerb, welcher Benutzer die beste WebExplorer-Animation erstellt. Die Ergebnisse und viele nette Animationen zum herunterladen und ausprobieren sind unter der WWW-URL des TeamOS/2 Österreich »http://www.os2forum.or.at/TeamOS2/Special/Animations/Contest« zu finden. Dort gibt es ein Archiv mit über hundert verschiedenen Animationen, vom »Java Duke« bis hin zur »NCC-1701D Enterprise«.



Abb. 9.50: Animationen für den WebExplorer

Diese Animationen werden vom WebExplorer automatisch eingebunden, sofern der Menüpunkt *Angepaßte Animationen* aktiviert ist.

Der Cache des IBM-WebExplorers

Der IBM-WebExplorer ist in der Lage, einen eigenen Cache für bereits geladene WWW-Seiten und Grafiken zu erstellen. Dieses wird unter dem Menüpunkt *Konfigurieren|Zwischenspeicherung* eingestellt.

Die Vorgabewerte von 16 Dokumenten und 32 Bilddateien sind sehr wenig. Wenn WWW-Seiten mit vielen Grafiken abgerufen werden, so sollte der Eintrag von 32 auf 256 erhöht werden (das Maximum). Dadurch wird der Abruf von der gleichen Seite beim Zurückblättern nach ein paar anderen Seiten enorm beschleunigt, weil diese Bilder ebenfalls noch im Cache sein können.

Die Anzahl der zwischenzuspeichernden Dokumente kann ebenfalls auf 256 erhöht werden.

Ob der Punkt *Zwischengespeicherte Abbilder im Hauptspeicher behalten* aktiviert werden soll oder nicht, hängt von den vorhandenen Hauptspeicherresourcen des Benutzers ab. Wird dieser Punkt aktiviert, so kann es recht schnell zu extensivem Swappen kommen und der Betrieb wird verlangsamt.



Abb. 9.51: Einbinden von Animationen

Sollen nur Dateien von einem lokal im Netzwerk betriebenen WebServer abgerufen werden, so kann das Caching auch komplett abgeschaltet werden, da der Datendurchsatz im lokalen Netz sicherlich groß genug sein wird, um dieses aufzufangen.

Die Benutzung eines Proxyservers mit dem IBM-WebExplorer

Sicherlich haben viele Benutzer schon einmal das Problem gehabt, daß Ihr Provider zwar einen Proxy-Server betreibt, dieser aber scheinbar mit dem IBM-WebExplorer nicht genutzt werden kann. Zur Frage, was ein Proxy-Server ist nur soviel: Es handelt sich dabei um ein Computersystem, auf dem ein Programm betrieben wird, das eine Datenbank über bereits von Benutzern abgerufene Internetdaten (ftp-Dateien, WWW-Seiten, etc.) führt. Diese bereits abgerufenen Daten werden auf diesem Computersystem ebenfalls zwischengespeichert. Die Daten können dann bei einer wiederholten Anfrage nach dieser Internetdatei direkt aus dem sogenannten Cache dieses Proxy-Servers mit einer wesentlich höheren Geschwindigkeit geliefert werden, anstatt diese erneut aus dem Internet zu laden.

Die Konfiguration des WebExplorers um einen Proxyserver zu benutzen ist sehr einfach, jedoch scheitert es bei vielen an einer winzigen Kleinigkeit: einem Slash »/«. Die Angabe muß im Server-Konfigurationsmenü folgendermaßen erfolgen: »http://proxy.adresse.des.providers-:port/«. Der abschließende Slash ist wichtig. Wird dieser vergessen, so kann keine URL mehr

angesprochen werden, solange der Punkt *Enable Proxy* aktiviert ist. Ist der Wert für Port gleich 80, so kann die Angabe *Port* auch komplett weggelassen werden. In einigen Fällen ist nach der Aktivierung eines Proxies ein Neustart des WebExplorers nötig.

Ein Problem mit dem WebExplorer und Proxy-Server-Caching besteht darin, daß der WebExplorer bei einem Refresh (einer erneuten Anfrage nach einer Seite um diese zu aktualisieren) diesen nicht korrekt formuliert und nur vom Proxy-Server die Seite erneut bekommt, diesen jedoch nicht veranläßt, das Dokument neu zu holen. Auf diese Art kann es passieren, daß ein Dokument aktuell erscheint, obwohl es eigentlich schon völlig veraltet sein könnte. Der Netscape Navigator führt diese Refresh-Anforderung korrekt durch und erhält immer die aktuellste Seite geliefert.

Verschiedenes zum WebExplorer

Mit dem WebExplorer ist der Benutzer auch in der Lage, Email zu versenden (jedoch nicht zu lesen), und er hat die Möglichkeit die Foren der Newsgroups im Internet/UseNet zu frequentieren und dort ebenfalls Informationen zu beziehen. Eine Beschreibung zu der Newsgroup-Funktionalität wird unter dem Kapitel über die Newsclients erfolgen, eine kurze Beschreibung zu der Fähigkeit Email zu versenden erfolgt unter dem Kapitel »Weltweit elektronisch Post versenden«.

Beim WebExplorer ist es möglich, URL-Objekte durch Drag&Drop recht einfach zu erstellen. Greift der Benutzer auf eine Web-Seite zu, die ihn interessiert und deren Adresse er gerne weiterhin gespeichert haben möchte, so zieht der Benutzer diese Seite einfach vom Fenster des IBM-WebExplorer auf die Arbeitsoberfläche oder in einen Ordner. Dadurch wird ein URL-Objekt erzeugt, das die Adresse dieser Seite als Einstellung gespeichert hat. Später ist der Benutzer dann in der Lage, durch einfaches Drag&Drop dieses URL-Objektes auf den Web-Explorer oder durch Doppelklick auf das Objekt diese Seite wieder anzusteuern.

Durch eine Kombination dieses URL-Objekts und URL-Ordner lassen sich recht komplexe und umfangreiche Quicklists oder Hotlists anlegen. Diese Ordner mit Unterordnern können zum Beispiel nach bestimmen Themen sortiert sein. Dort in diese Ordner werden dann entsprechend die den Themen zugeordneten URL-Objekt einsortiert und schon ist der Quicklist-Ordner fertig.

Natürlich können nicht nur URL-Objekt erzeugt werden. Zum Beispiel ist es auch möglich, ein Bild, das auf einer Seite angezeigt wird, einfach per Drag&Drop auf ein Laufwerk oder die Arbeitsoberfläche zu sichern, um dieses so zu speichern (für eine eventuelle Weiterverarbeitung). Andersrum können natürlich auch Bilder in einem Format, das vom WebExplorer unterstützt wird, einfach per Drag&Drop auf den WebExplorer fallen gelassen und dadurch angezeigt werden.

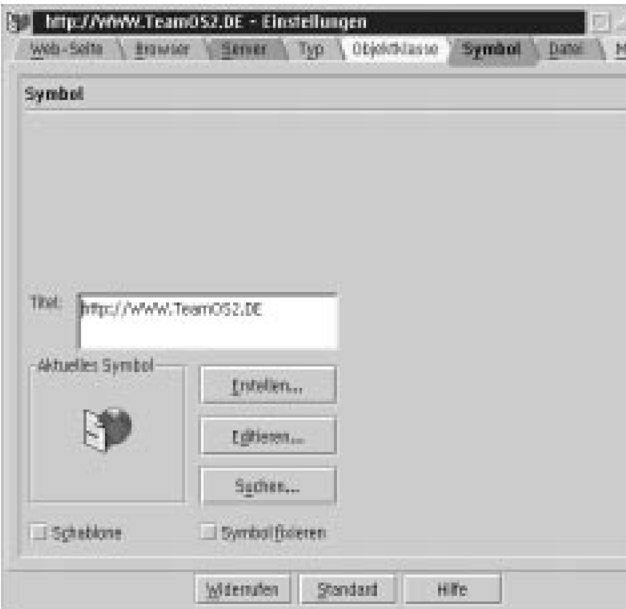


Abb. 9.52: WWW-URL-Objekt der TeamOS/2 Deutschland-WWW-Seite



Abb. 9.53: Quicklist-Ordner

9.10.4 Das Erdmännchen im Internet: Gopher

Bevor Informationen über das WorldWideWeb (oder wie es mittlerweile auch von Personen liebvoll bezeichnet wird: World Wide Wait (weltweites warten, das ein Anspiel auf die viel zu kleinen Kapazitäten der Datenleitungen in Deutschland darstellen soll) verteilt wurden, existierte schon ein hypertextähnliches System im Internet, das ebenfalls Informationen öffentlich bereitstellte.

Anders als im WorldWideWeb wurden diese Informationen nicht multimedial mit Grafiken, Sounds oder wie neuerdings mit Javaprogrammen aufbereitet, sondern auf der Basis eines textorientierten Browsers zur Verfügung gestellt. Dadurch ist die Übertragung der Daten-, oder wie hier genannt, Menüseiten, wesentlich schneller, da die übertragene Datenmenge wesentlich kleiner sein kann. Natürlich gibt es auch im WorldWideWeb die Möglichkeit, das Laden der Grafiken zu deaktivieren, was als Ergebnis eine ähnliche Geschwindigkeit erreicht, wo jedoch unter Umständen die Hälfte der Informationen verloren gehen kann.

Die Informationen selber können natürlich auch weiterhin Grafiken, Sounddateien oder Datendateien sein, diese sind selbstverständlich auch über dieses System abrufbar. Für den Abruf von Daten über dieses sogenannte Gophersystem ist ein Gopher-Client nötig. Dieser lädt die Informationen vom Gopherserver und bereitet sie auch auf. Auch ist es mit Gopher möglich, den Dienst ftp zu nutzen. Dieses verläuft ähnlich wie der Transfer mit einen WWW-Client.

Gopher heißt übersetzt Erdmännchen. Von einigen Anwendern wird Gopher allerdings fälschlicherweise auch mit Maulwurf oder Biber übersetzt. Wie dieser Name entstanden ist, kann nur geraten werden.

Die über das Gophersystem verbreiteten Informationen sind hirarchisch angeordnet. Es gibt eine Art »Unterverzeichnis« (Typ 1), das auf weitere Informationen zu einem Gebiet verzweigt. Natürlich gibt es auch Dokumente oder Dateien (Typ 0) oder sogar Programme (variierender Typ, 7 zum Beispiel), die auf dem Gopherserver (wartet auf Port TCP 70) ausgeführt werden können (Datenbanken zum Beispiel).

Zu den unterschiedlichen Dateien, die über Gopher gesammelt werden können, lassen sich auch Zuordnungen zu den dafür verwendeten Programmen treffen, um diese dann entsprechend weiter zu verarbeiten.

Es gibt trotz WWW im Internet noch sehr viele Gopherserver. Bei Warp V4 wird ein Gopherclient mitgeliefert, der auf dem PM aufbaut. Wird der Gopherclient unter OS/2 gestartet, so erscheint das standardmäßig konfigurierte Gopherstartmenü. Beim ersten Start von Gopher zeigt das Gophermenü den Inhalt des Startmenüs des Gopherservers »os2info.gopher.ibm.com« der IBM in den USA.

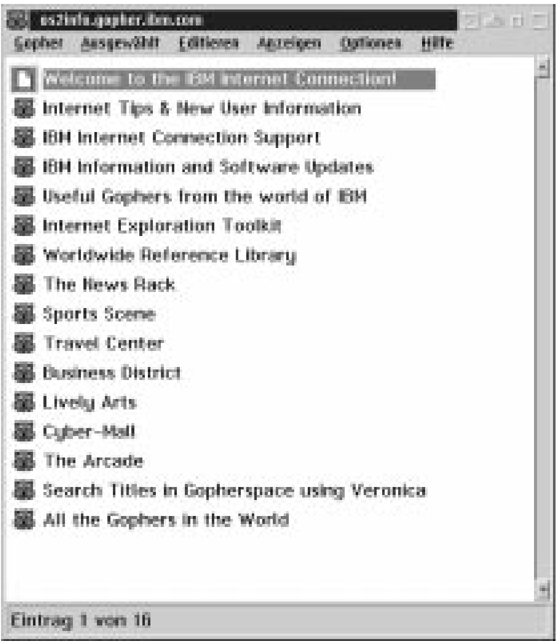


Abb. 9.54:
Das Standardstartmenü



Abb. 9.55: All the Gophers in the World

Natürlich ist es möglich zu konfigurieren, welchen Ursprungspunkt dieses Startmenü ansteuern soll. Dabei wird einfach die entsprechende Menüseite ausgewählt und über den Menüpunkt *Gopher* der Punkt *Dieses Menü als Ausgangslesezeichen definieren* angewählt. Danach wird bei jedem Programmstart von Gopher genau dieses Informationsmenü geladen.

Trotz der weiten Verbreitung des WorldWideWeb ist es keineswegs der Fall, daß die Informationen, die über Gopher abrufbar sind, völlig veraltet sind. Diese Daten werden größtenteils gepflegt und ständig aktualisiert (einige Informationen im Internet sind schließlich auch völlig veraltet ...).

9.10.5 FileTransfer: Dateien kopieren leicht gemacht

Mit Warp V4 werden eine ganze Reihe von Möglichkeiten mitgeliefert, mit denen der Benutzer in der Lage ist, über ein auf dem TCP/IP-Protokoll basierenden Netzwerk Dateien auf das lokale Computersystem zu kopieren. Dieser Vorgang beruht auf dem im Internet weit verbreiteten ftp-Dienst. *ftp* bedeutet *»File Transfer Protocol«* und wörtlich übersetzt etwas wie »Dateiübertragungsprotokoll«. Dieser Begriff beschreibt den Vorgang schon recht gut.

Bei Warp V4 sind eine ganze Reihe von Dienstprogrammen mitgeliefert, mit denen der Benutzer auf den ftp- oder tftp-Dienst zugreifen kann, darüberhinaus existieren auch spezielle ftp-Objekte der Workplace Shell.

ftp von der OS/2-Kommandozeile im Textmodus

Bei dem Dienstprogramm ftp.exe handelt es sich um ein Dienstprogramm, mit dem der Benutzer in der Lage ist, den Dienst ftp auf dem lokalen OS/2-Warp-Computersystem im Textmodus auszuführen. Das Programm wird durch den einfachen Aufruf »ftp« von der Kommandozeile gestartet. Nachdem das Programm gestartet wurde, befindet sich der Benutzer an einer Art Befehlsprompt, über den die weitere Bedienung des Programms möglich ist. Durch die Eingabe von »help« oder »?« wird eine Liste mit den zulässigen Befehlen generiert. Durch die entsprechende Eingabe von »help <*Kommando*<« wird der gewählte Befehl näher erläutert. Die meisten Befehle werden jedoch nur selten im normalen Gebrauch benutzt.

An dieser Stelle werden nur die am häufigsten verwendeten Befehle, die in verschiedene Kategorien unterteilt werden können, näher erläutert. Der erste Teil ist die Gruppe der verbindungsrelevanten Anweisungen, die folgende Gruppe erläutert die transferrelevaten Anweisungen und in der letzten Kategorie werden nützliche Befehle erläutert.

Verbindungsrelevante ftp-Befehle

open, close, disconnect, bye, quit, user

Im folgenden wird näher auf die Anweisungen eingegangen, mit denen es möglich ist, eine Verbindung zu einem entfernten Computersystem über ftp aufzubauen, sich einzuloggen und die Verbindung wieder zu beenden.

Der Befehl *open* startet den Verbindungsaufbau zu einem Computersystem, dessen Namen an den Befehl als Parameter angehängt wird, oder das der Benutzer bei einer Nachfrage eingibt.

Diese Anweisung ist gleichbedeutend mit dem Aufruf von »ftp.exe« gefolgt durch den Computersystemnamen, auf den der Zugriff erfolgen soll.

Konnte die Anweisung erfolgreich ausgeführt werden, besteht nun eine Verbindung zu einem Computersystem (auch »ftp-Server« genannt) über das IP-Netzwerk oder zu dem eigenen lokalen Computersystem, wenn dieses angesprochen wurde. Der ftp-Daemon-Prozeß wartet auf eine Verbindungsanforderung auf Port TCP 21. Werden Dateien übertragen, so erfolgt die Übertragung der Daten über den Port TCP 20 der Serverseite.

Es erscheint bei einem erfolgreichen Verbindungsaufbau in der Regel ein kleiner Begrüßungsschirm und eine Aufforderung, einen Benutzernamen zu spezifizieren. Diese Anfrage wird normalerweise von einer Anfrage nach einem Passwort zur Authentifizierung gefolgt. Gibt der Benutzer an dieser Stelle einen falschen Benutzernamen oder ein falsches Passwort ein oder vertippt sich, ist es möglich, später mit dem Befehl »user *<username<*« und erneut folgender Passwortabfrage diesen Vorgang zu wiederholen.

Ist dieser Prozeß erfolgreich abgelaufen, wird eine entsprechende Nachricht mit der Bestätigung der erfolgreichen Verbindungsaufnahme ausgegeben. Sehr viele ftp-Server im Internet sind für alle Benutzer des Internets verfügbar. Es existiert eine einheitliche Regelung, um auf solche Computersysteme zuzugreifen. Der Loginprozeß erfolgt über einen virtuellen Benutzer mit der Kennung *anonymous* oder *ftp*. Für das Passwort wird erwartet, daß die eigene Email-Adresse eingegeben wird. In diesem Fall wird auch von »anonymous-ftp« gesprochen, »anonymer Dateitransfer«, weil der Verbindungspartner kein eigentlicher Benutzer des Systems ist, sondern eher ein »Unbekannter«.

Mit den Befehlen *close* oder *disconnect* und *bye* oder *quit* kann nun die Verbindung beendet werden, wobei der Gebrauch von *bye* oder *quit* auch das Programm ftp beendet. Möchte man jedoch nur die aktuelle Verbindung beenden und mit einem anderen Computersystem danach über die Anweisung *open* eine weitere Verbindung eingehen, so wird *close* oder *disconnect* verwendet, wobei lediglich die Verbindung zum bestehenden Computersystem beendet, nicht jedoch das Programm verlassen wird.

Transferrelevante ftp-Befehle

binary, image, ascii, get, recv, mget, put, mput, cr

Es gibt im ftp-Bereich verschiendene Möglichkeiten, eine Datei zu transferieren. Man kann zwischen »Binärdateien« und »ASCII-Dateien« unterscheiden. Soll eine Binärdatei übertragen werden, sollte auf dem Kommandoprompt des ftp-Programms der Befehl *binary* oder *image* eingegeben werden. Mit diesem Befehl wird ftp mitgeteilt, daß eine Binärdatei transferiert werden soll. Mittlerweile sind jedoch viele ftp-Server so konfiguriert, daß die Übertragung standardmäßig im Binärmodus erfolgt.

Werden nur ASCII-Dateien übertragen, so kann über den Befehl *ascii* auf dieses Format umgestellt werden.

An dieser Stelle sei dennoch eine kleine Warnung ausgesprochen. Die häufigste Fehlerursache für defekt übertragene Binärdateien (Zip-Archive zum Beispiel) liegt darin, daß als Format »ASCII« gewählt wurde, während eine solche Übertragung zwingend das Binär-Format erfordert.

Wurde sichergestellt, daß das richtige Format zum Übertragen benutzt wird, so kann jetzt die Übertragung einer Datei mit den Befehlen *get*, *recv* oder *mget* von dem entfernten Computersystem auf das lokale System erfolgen. Mit dem Befehl *get* oder *recv* wird **exakt eine Datei** transferiert, deren Name zudem auch genau bekannt sein muß. Da im Internet die meisten Computersysteme auf der Betriebssytemplattform Unix basieren, das fast immer zwischen groß- und kleingeschriebenen Zeichen unterscheidet, muß exakt diese Schreibweise für den Dateinamen benutzt werden.

Wird jedoch statt *get* die Anweisung *mget* benutzt, können auch Dateinamen angegeben werden, die durch Platzhalter (Wildcards) verkürzt werden. Der gebräuchlichste Platzhalter ist das Zeichen »*«. Durch den Aufruf von »get *<dateiname> <mein_name>*« wird die Datei »<dateiname>« vom entfernten Computersystem auf das lokale System übertragen und unter dem Namen »<mein_name>« gespeichert. Diese Funktionalität ist sehr nützlich, wenn versucht wird, Dateien mit langen Dateinamen auf das archaische und veraltete FAT-Filesystem zu speichern. In den meisten Fällen erhält der Benutzer sonst abgehackte Namen. So wird jedoch ermöglicht, die Dateien unter einem hinterher im begrenzten Maße verständlichen Namen zu speichern.

Sollen Dateien von dem lokalen Computersystem auf ein entferntes System übertragen werden, so müssen die Befehle *put* und *mput* dafür verwendet werden. Die Handhabung ist identisch mit der Benutzung von *get* und *mget*, das heißt, nur bei *mput* dürfen Platzhalter verwendet werden, während bei *put* der genaue Dateiname bekannt sein muß. Durch Angabe von »put *<lokaler_dateiname> <entfernter_dateiname>*« kann ebenfalls eine Umbenennung der übertragenen Datei stattfinden.

Sollen Dateien im ASCII-Format von einem Unix-Computersystem auf das lokale Computersystem oder umgekehrt übertragen werden, so besteht die Möglichkeit, durch die Angabe des ftp-Befehls *cr* die Konvertierung von CR/LF an- bzw. abzuschalten. Dadurch erscheinen in den Textdateien auf den Zielsystemen keine störenden Steuerzeichen an den Zeilenenden.

Nützliche und andere grundlegende ftp-Befehle

Prinzipiell gehören alle nicht bereits oben aufgeführten ftp-Befehle in diese Kategorie. Wird auf der ftp-Kommandozeile die Anweisung *help* eingegeben, so ist ersichtlich, daß die Anzahl der ftp-Befehle wesentlich größer ist als die hier aufgeführten. Die wichtigsten nützlichen Anweisungen, die hier jetzt behandelt werden sind *!*, *bell*, *cd*, *cdup*, *delete*, *dir*, *hash*, *keepdate*, *lcd*, *ls*, *mdelete*, *mkdir*, *prompt*, *pwd*, *rename*, *rmdir*, *status* und *verbose*.

Durch die Eingabe von *!* gefolgt von [Enter] wird auf eine normale OS/2-Kommandozeile gesprungen, in der ganz normale Tätigkeiten ausgeführt werden können. Durch die Eingabe von »exit« in dieser Kommandozeile gelangt der Benutzer wieder auf den ftp-Kommandoprompt zurück. Durch die Eingabe von »!*<OS/2-Befehl>*« kann auch ein *<OS/2-Befehl>* ausgeführt werden, ohne die ftp-Kommandozeile zu verlassen.

Benutzer, die gerne auch ein akustisches Signal bei erfolgreicher Vollendung eines ftp-Befehls wünschen, können durch die Eingabe von *bell* eine entsprechende Funktion aktivieren bzw. deaktivieren. Dadurch ertönt nach jeder vollendeten Anweisung ein Piepston aus dem Lautsprecher des lokalen Computersystems.

Durch die Befehle *cd*, *cdup*, *lcd* und *pwd* kann in verschiedene existierende Unterverzeichnisse gewechselt werden, wobei *pwd* das aktuelle Verzeichnis auf dem entfernten Computersystem ausgibt. Durch *lcd* kann auf dem lokalen Computersystem ein Verzeichniswechsel durchgeführt werden, mit *cd* auf dem entfernten Computersystem. Mit dem Befehl *cdup* wird auf dem entfernten System in das übergeordnete Verzeichnis gewechselt.

Mit den Befehlen *mkdir* und *rmdir* können neue Verzeichnisse angelegt, bzw. vorhandene leere Verzeichnisse auf dem entfernten Computersystem gelöscht werden. Im Zusammenhang mit *!* kann dieses dann auch für das lokale Computersystem erfolgen.

Die Befehle *dir* und *ls* (mit einigen möglichen Unix-ls-Parametern) ermöglichen es, den Inhalt von Verzeichnissen auszugeben. Durch die Eingabe von »dir *<Verzeichnis>*« wird der Inhalt des Verzeichnisses »<Verzeichnis>« auf dem Bildschirm ausgegeben. Um das komplette Verzeichnis mit allen Unterverzeichnissen und Dateien anzeigen zu lassen, kann der Befehl *ls -Rals* benutzt werden. Jedoch unterstützt nicht jeder ftp-Server diese Funktionalität. Abhängig von dem Angebot des ftp-Servers kann dieser Befehl unter Umständen eine große Menge an Daten übertragen. Wird dieses nicht in eine Datei geleitet, so ist es im Normalfall nicht sinnvoll möglich, diese Ausgabe zu verwerten. Die Umleitung in eine Datei erfolgt durch die Angabe eines weiteren Parameters, der dem lokalen Dateinamen entspricht.

Dateien, die versehentlich übertragen wurden, können durch die Befehle *delete* oder *mdelete* (mit Platzhaltern) gelöscht werden. Wenn der Transfer einer Datei auf einen ftp-Server aus den verschiedensten Gründen fehlgeschlagen ist und der Befehl *delete* nicht erlaubt ist, kann versucht werden, die Datei mit dem *rename*-Befehl umzubennen und den Transfer erneut zu starten, um auf dem Zielsystem den Originalnamen für diese Datei beizubehalten.

Es besteht die Möglichkeit, den TimeStamp (also das Erstellungsdatum und die Erstellungszeit) der Originaldatei über die Angabe der Anweisung *keepdate* beizubehalten, um so einen Überblick zu behalten, wann diese Datei auf dem ftp-Server erstellt wurde.

Über die Eingabe des Befehls *hash* kann zwischen zwei Anzeigen umgeschaltet werden, die während der Übertragung einer Datei den Fortschritt des Transfers anzeigen. Standardmäßig ist die Anzeige der übertragenen Bytes eingeschaltet (ein fortlaufender Bytecounter), durch Eingabe von *hash* wird auf eine alternative Anzeige umgeschaltet. Dann werden Hash-Kreuzchen (»#«) angezeigt. Normalerweise wird ein Hash-Kreuz pro x transferierten Kilobyte der Datei dargestellt. Die Anzahl variiert zwischen den unterschiedlichen ftp-Servern. In den meisten Fällen wird die Ausgabe auf eine »1# pro 1kb«-Anzeige geschaltet.

Wenn Operationen mit vielen Dateien unternommen werden sollen und mit den Kommandos *mput*, *mget* und *mdel* gearbeitet wird, so kann mit der Eingabe von *prompt* der interaktive Modus abgeschaltet bzw. eingeschaltet werden. Im interaktiven Modus wird für jede Dateioperation eine Bestätigung angefordert, was bei der Ausführung von einem *mget* * recht schnell störend wirken kann. Wird der interaktive Modus ausgeschaltet, werden sämtliche Operationen ohne eine Rückfrage (außer im Fehler- oder in einem besonderen Fall) durchgeführt (Vorsicht sei geraten bei Operationen wie zum Beispiel *mdelete* *!).

Wird eine Verbindung zu einem ftp-Server aufgebaut, findet eine recht ausführliche Ausgabe beim Login und unter Umständen auch bei den Verzeichniswechseln statt. Soll diese Ausgabe

aber unterdrückt werden, um den Ablauf zu beschleunigen oder Bandbreite zu sparen, so kann dieses durch die Eingabe des Befehls *verbose* erfolgen.

Um eine Übersicht über die derzeit konfigurierten Parameter des ftp-Programms zu erhalten, kann der Befehl *status* benutzt werden. Dort werden die diversen Konfigurationsmöglichkeiten angezeigt.

ftp-Vorgänge automatisieren, *NetRC*, *Makros* und Kommandozeilenparameter

Es besteht die Möglichkeit, über eine Konfigurationsdatei Einstellungen, die sonst per Hand jedes Mal erneut über die ftp-Kommandozeile eingegeben werden müßten, automatisch vornehmen zu lassen. Ebenfalls besteht auch die Möglichkeit, bestimmte Abläufe zu automatisieren, wenn auf einen spezifischen ftp-Server zugegriffen wird. An dieser Stelle sei auf die Online-Dokumentation von Warp V4 verwiesen, in der dieses erläutert wird (wobei das Finden der entsprechenden Seiten ein wenig dauern kann: *Unterstützung|Information|Funktionen|TCP/IP Benutzerhandbuch|Dateien zwischen Hosts übertragen*).

Eine Umgebungsvariable »set netrc=<laufwerk>:\pfad\meine_net.rc« wird in die »config.sys« eingetragen und wird somit erst beim nächsten Systemstart aktiv. In diese Datei »NetRC« trägt der Benutzer nun nach folgendem Beispiel-Muster die Einstellungen ein, die beim Aufruf von ftp aktiviert werden sollen, wobei jeweils die Konfiguration für ein Computersystem mit dem Begriff *machine* begonnen werden muß:

```
machine hobbes.nmsu.edu login anonymous password meine@Email.adresse macdef
mymacro
binary
bell
prompt
keepdate
sunique
runique
cd incoming
ls -als

default macdef init
binary
bell
prompt
keepdate
sunique
runique
```

Startet man jetzt das ftp-Dienstprogramm, geschieht erstmal noch nichts. Wird jetzt eine Verbindung zu einem ftp-Server aufgebaut (durch »open <Computersystemname>« oder durch übergeben eines Parameters bei dem Aufruf des Programms), so werden nach der Loginprozedur die Befehle nach *default macdef init* ausgeführt.

Wird eine Verbindung zu dem bekanntesten OS/2-Software führenden ftp-Server »hobbes.nmsu.edu« in den USA aufgebaut, so wird automatisch ein Login als User *anonymous* mit der Email-Adresse als Password durchgeführt.

Die jetzt unter *default macdef init* eingetragenen Befehle werden nicht ausgeführt. Dafür kann jetzt jedoch unter der Eingabe der Anweisung *$* für Makroausführung unter Verwendung von *mymacro* als Parameter das in der »NetRC« definierte Makro für diesen Server ausgeführt werden.

Auf diese Weise ist es möglich, für jeden benutzten ftp-Server ein kleines Startskript zu erschaffen.

Oben wurde ein Makro mit Namen »mymacro« definiert. Es ist so möglich, beliebige Makros zu erstellen und diese dann während der ftp-Benutzung durch »*$*MakroName« auszuführen. Die Art der Makros kann von simplen Verzeichniswechseln über Dateitransfer bis hin zu wiederholten Anweisungen gehen. Diese Programmierung wird in der Online-Hilfe von Warp V4 näher erläutert.

Das Dienstprogramm ftp kann auch mit Kommandozeilenparametern gesteuert werden, wobei die Übergabe des Namens des ftp-Servers nur ein möglicher Parameter davon ist. Wenn ftp mit der Option *-n* aufgerufen wird, wird ein eventuell vorhandenes NetRC-File ignoriert und eventuell dort enthaltene Befehle werden nicht ausgeführt.

Wird ftp mit der Option *-i* gestartet, findet kein interaktiver Ablauf statt. Es besteht die Möglichkeit, ein sogenanntes ftp-Skript zu erstellen, das ähnlich der »NetRC« einen Skriptablauf ermöglicht. Dabei werden alle ftp-Befehle in diese Datei geschrieben und ftp durch den Aufruf »ftp -i < <skriptname>« gestartet. Um ganz sicher zu gehen, daß auch wirklich nur die Daten in dem ftp-Skript benutzt werden sollen und keine Angaben aus der »NetRC«, erfolgt der Aufruf über »ftp -n -i < <skriptname>«.

ftp made by PM

Das Objekt ftp-PM

Das Starten des Dienstprogramms ftp-PM erfolgt entweder über einen Doppelklick auf das ftp-PM-Objekt (oder auf eine Referenz oder eine Kopie des Objektes) oder durch Aufruf von »ftppm.exe« auf einer OS/2-Kommandozeile. ftp-PM meldet sich mit einer Informationsmaske, in der Angaben zu dem *gewünschten ftp-Server* und der *Benutzerkennung* auf diesem Computersystem gefordert werden. Die Angabe zu *Konto* kann leer bleiben. Nach Eingabe dieser Daten (hier als Beispiel: *ftp-Server-Name* ist »ftp.TeamOS2.de«, der *Login* geschieht durch *anonymous-ftp)* erscheint das Arbeitsfenster des ftp-PM.

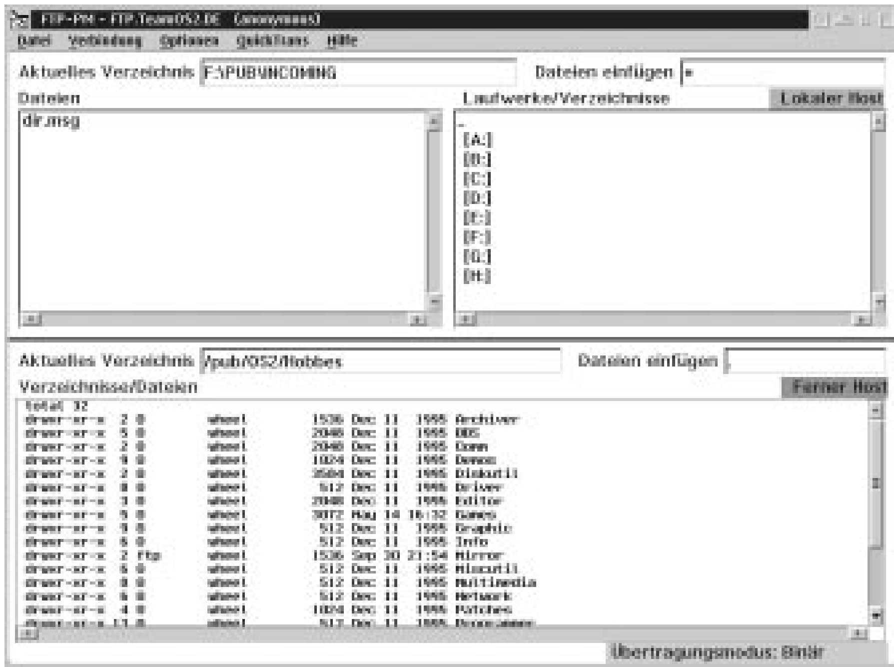


Abb. 9.56: ftp-PM-Arbeitsfenster

Das Wechseln in eines der angezeigten Unterverzeichnisse auf dem ftp-Server (unterer Teilbereich des ftp-PM-Fensters) erfolgt durch einen Doppelklick auf den Namen des Verzeichnisses oder durch eine entsprechende Eingabe bei der Pfadangabe. Das Zurückspringen in das übergeordnete Verzeichnis erfolgt durch die Korrektur der Eingabe bei *Aktuelles Verzeichnis*. Wird bei der *Dateien einfügen*-Maske ein Schlüssel angegeben (z.B. **.txt*), so werden nur Dateien angezeigt, die diesem Kriterium entsprechen.

Die Übertragung von Dateien erfolgt entweder durch das Auswählen der Dateien mit der Maus und dem Aktivieren von *Quicktrans* oder per *Drag&Drop* des *selektierten Bereichs* in den oberen linken Teil des ftp-PM-Arbeitsfensters. Je nach Konfiguration des ftp-PM erscheint nun ein Fenster mit einer Aufforderung, ob diese Datei wirklich übertragen werden soll und wie der lokale Dateiname sein soll. Das Transferieren von Dateien von dem lokalen Computersystem auf den ftp-Server kann auf die selbe Art und Weise geschehen. Dazu werden die markierten Dateien aus dem oberen linken Bereich in den unteren Bereich gezogen.

Die *Bestätigungsaufforderung* kann *deaktiviert* werden, indem der Menüpunkt *Optionen Bestätigung* auf *Nein* geschaltet wird. Der Transfer beginnt dann sobald das Drag&Drop-Manöver abgeschlossen wurde. Die Dateinamen, unter denen diese Dateien gespeichert werden, sind mit den Ursprungsnamen identisch, sofern das Filesystem, auf dem diese Dateien gespeichert werden sollen, dieses zuläßt (Stichwort 8.3 und FAT). Zu beachten ist hier auch

wieder der Transfermodus. Standardmäßig ist dieser auf *ASCII* eingestellt, ein Umstellen auf BINARY empfiehlt sich über *Optionen Übertragungsmodus*.

Möchte der Benutzer ein ganzes Unterverzeichnis downloaden (von einem ftp-Server herunterladen, kopieren), so genügt es, das Unterverzeichnis zu selektieren und per Drag&Drop zu verschieben, wie es mit den Dateien vorher geschehen ist. Allerdings ist es auf diese Weise nicht möglich, Unterverzeichnisse dieses Verzeichnisses mit zu kopieren. Selbst wenn die entsprechenden Verzeichnisse auf dem Zielcomputersystem angelegt sind, scheitert dieser Versuch.

Störend ist, daß ftp-PM nicht in der Lage ist, die Einstellungen, die getätigt wurden, zu speichern. Wird ftp-PM das nächste Mal gestartet, so sind die gleichen Standardeinstellungen vorhanden wie beim ersten Start. Um dennoch zu erreichen, daß einige Änderungen dauerhaft sind, kann natürlich die Einstellung des ftp-PM-Objektes geändert werden.

ftp-PM für verschiedene ftp-Server konfigurieren

Es existieren für ftp-PM zwei verschiedene Schablonen (abhängig von der Art des Netzwerkzuganges, entweder über das LAN oder über Modem), mit denen ftp-PM konfiguriert werden kann. Um ein neues Objekt für ftp-PM zu erstellen, wird einfach ein Programmobjekt vom Schablonenobjekt abgezogen.

In den Einstellungen des Objektes werden auf der Seite *Host* die Daten (Adresse) für den gewünschten ftp-Server und die Zugangsdaten (Benutzerkennung) unter der Seite *Identifikation* eingestellt. Auf der Seite *Optionen* kann festgelegt werden, ob der Transfer im Binär- oder ASCII-Modus stattfinden soll, ob eine Bestätigung für den Transfer gewünscht ist und ob einzigartige Dateinamen vergeben werden sollen (wenn eine Datei gleichen Namens bereits vorhanden ist wird an den Namen eine Zahl angehängt).

Ferner ist es in den Einstellungen auch möglich, ein Startverzeichnis auf dem ftp-Server anzugeben, in dem sich der Benutzer befindet, wenn ftp-PM die Verbindung zum ftp-Server hergestellt hat (sofern es dieses Verzeichnis gibt und der Benutzer Zugriffsrechte darauf besitzt). Dieses kann auch für das lokale System eingestellt werden. Dann befindet sich der Benutzer auf dem gewünschten lokalen Laufwerksverzeichnis, auf das die übertragenen Dateien gespeichert werden sollen.

Existiert eine gültige »NetRC«, so wird diese von ftp-PM ebenfalls benutzt, wenn auf einen entsprechend in der »NetRC« definierten ftp-Server zugegriffen werden soll.

Das Löschen von lokalen Dateien oder von Dateien auf dem ftp-Server (entsprechende Zugriffsrechte vorrausgesetzt) kann über das Menü erfolgen, oder es ist ebenfalls möglich, die Dateien zu markieren und dann über die [Entf]-Taste diese Dateien zu löschen.

tftp, triviales ftp

Der Unterschied zwischen dem normalen ftp-Dienstprogramm »ftp.exe« und dem Dienstprogramm »tftp.exe« ist recht einfach ersichtlich, wenn der Benutzer das Dienstprogramm »tftp.exe« auf einer OS/2-Kommandozeile startet und mit *?* die Befehlsübersicht anfordert. Die Anzahl der bei tftp vorhandenen Anweisungen ist sehr viel geringer als bei ftp.

Bei tftp ist es nicht möglich, mehrere Dateien auf einmal zu transferieren. Es ist möglich, immer nur eine einzelne, genau bekannte Datei zu übertragen. Es ist auch nicht möglich, in andere Verzeichnisse zu wechseln oder den Inhalt eines Verzeichnisses anzuzeigen.

Ein weiterer Unterschied besteht in der Art der Zugriffsautentifizierung. Es wird keine konventionelle Autentifizierung über einen Benutzernamen und ein Passwort hergestellt. Wenn eine Verbindung zu einem tftp-Server aufgebaut wird, kann mit dem Dateitransfer prinzipiell sofort gestartet werden, sofern das verwendete lokale Computersystem über eine gültige Zugriffserlaubnis auf dem entfernten Serversystem verfügt. Nur aufgrund des die tftp-Verbindung aufbauenden Computersystems und dessen Computersystemnamens kann die Authentifizierung erfolgen.

Ähnlich wie bei ftp kann auch hier der Transfermodus eingestellt werden, der für die entsprechende Datei benötigt wird (*ASCII* oder *BINARY*).

Das Kopieren oder Downloaden der Dateien geschieht über den Befehl »get <dateiname>«, wobei <*Dateiname*> exakt bekannt sein muß. Nur wenn auf dem tftp-Server diese Datei in dem Verzeichnis liegt, das diesem Computersystem zugeordnet ist, kann ein Transfer erfolgen. Ist die Datei nicht vorhanden, wird lediglich eine Fehlermeldung ausgegeben.

Solch ein Verfahren ohne interaktive Autentifizierung durch einen Benutzernamen und ein Passwort kann dazu verwendet werden, um bestimmte Computer-Netzwerksysteme über ein TCP/IP-Netzwerk zu starten (z.B. einen Livingston Portmaster) oder auf schnelle Art einen Abgleich einer bestimmten bekannten Datei zu tätigen.

Der ftp-Server auf dem lokalen Computersystem

Mit Warp V4 wird ein spezielles Daemonprogramm (»ftpd.exe«) mitgeliefert, das die Möglichkeit bietet, einen sogenannten ftp-Server-Daemon auf dem lokalen Computersystem zu starten. So kann ermöglicht werden, daß ftp-Zugriffe von ausserhalb des lokalen Systems auf Dateien dieses Systems erfolgen kann.

Das Starten des ftpd

Die Einrichtung und Konfiguration dieses Dienstprogramms erfolgt über das Konfigurationsnotebook des TCP/IP-KonfigurationsProgramms (im Ordner *Systemkonfiguration* befindet sich das TCP/IP-Konfigurationsobjekt). Die Konfiguration der Zugriffsberechtigungen kann auch per Hand in der Datei »trusers« (steht für *trusted users*, vertraute Benutzer) im »etc«-Verzeichnis des »mptn«-Verzeichnisses geschehen.

Um den ftp-Server zu starten, gibt es drei generelle Möglichkeiten. Die erste besteht darin, auf der OS/2-Kommandozeile »ftpd.exe« einzugeben. Der Serverprozeß wird dadurch gestartet und es sind nun Logins über das Dienstprogramm ftp möglich. Jedoch müssen noch, wie oben beschrieben, entsprechende Zugriffsrechte und Benutzer konfiguriert und angelegt werden.

Die zweite und dritte Möglichkeit besteht darin, den ftpd-Prozess automatisch beim Systemstart aktivieren zu lassen. Entweder über den inetd-Superserverprozess oder über das Startskript »tcpstart.cmd«, das auch den inetd-Server-Prozess startet. Es besteht noch die Auswahl zwischen *Hintergrundsitzung* und *Vordergrundsitzung*, wenn der ftpd-Prozess nicht über den

inetd gestartet werden soll.

Bei der Auswahl von *Hintergrundsitzung* wird ftpd so gestartet, daß *keine Ausgabe* auf dem Bildschirm zu sehen ist und das Programm erscheint nicht in der Fensterliste. Bei *Vordergrundsitzung* erscheint jedoch ein OS/2-Fenster in der Fensterliste und auf dem Bildschirm. So ist es möglich, den Ablauf von ftp-Verbindungen zu sehen. Stört jedoch das Fenster des ftpd beim Systemstart, so kann die Option *minimiert* aktiviert werden und das Fenster bleibt versteckt, bis es über die Fensterliste aufgerufen wird.

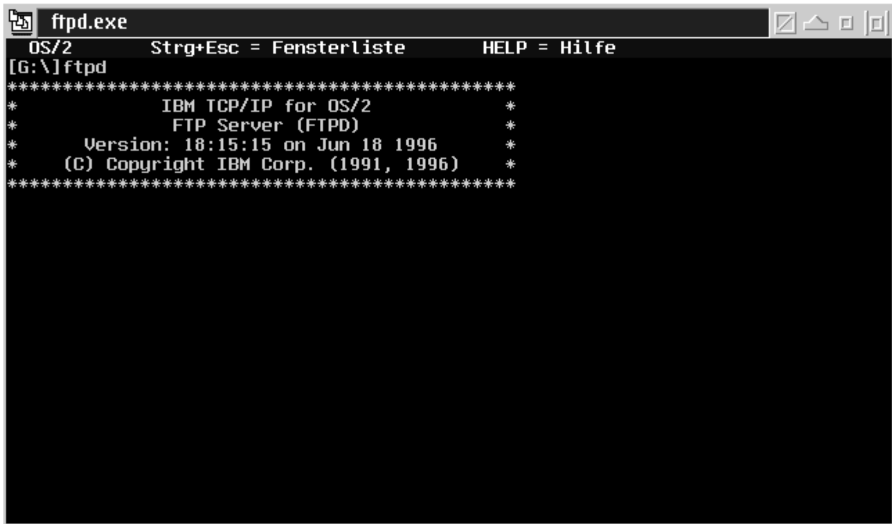


Abb. 9.57: Der ftpd-Serverprozeß

Konfiguration von ftp-Benutzern

Es existiert eine Datei mit Namen »TRUSERS« im »etc«-Verzeichnis des »mptn«-Verzeichnisses, wobei diese beim ersten Start des OS/2-Systems noch nicht vorhanden ist. Diese Datei wird erst angelegt, wenn eine Eingabe über das TCPIP-Konfigurationsprogramm auf der Seite 1 der Sicherheitseinstellungen im Bereich *ftp-Zugriff* geschehen ist. Auf dem Screenshot ist bereits der Benutzer *anonymous* eingetragen.



Abb. 9.58: Benutzerkonfiguration über das TCP/IP-Konfigurationsprogramm

Betrachtet man die Datei »trusers«, so ist der Aufbau leicht erkenntlich und recht einfach strukturiert.

```
user: <benutzername> <password>
rd: f:\pub
wr: f:\pub\incoming
default: f:\pub
loginfo: logall
```

Durch die Zeile »user: <benutzername> <password>« wird der Benutzer mit dazugehörigem Passwort festgelegt. Handelt es sich um den Benutzer *anonymous*, so kann das Passwort weggelassen werden und der OS/2-ftpd fragt nicht nach einem Password bei einem Verbindungsaufbau dieses Benutzers. Wird ein beliebiges Wort für das Passwort in der Zeile angegeben, verlangt der OS/2-ftpd nach einem Passwort beim Verbindungsaufbau und es kann ein beliebiges Passwort (meist die Email-Adresse des Benutzers) angegeben werden. *anonymous* ist der einzige Benutzer, dem kein Passwort zugeordnet werden muß. Für alle anderen Benutzer muß ein Passwort konfiguriert werden, sonst erfolgt kein korrekter Verbindungsaufbau.

```
rd: f:\pub
wr: f:\pub\incoming
default: f:\pub
```

Diese Zeilen spezifizieren den Zugriff auf die verschiedenen Verzeichnisse des ftp-Servers. *rd* bedeutet *read directory*, also die Verzeichnisse, die für den Nur-Lese-Zugriff freigegeben sind. Alles unter f:\pub kann ebenfalls gelesen werden. Andere Verzeichnisse können nicht gelesen

werden. Steht jedoch ^*rd:* dort, so ist der Lesezugriff auf alle Verzeichnisse freigeschaltet, nur für das hinter ^*rd:* stehende nicht.

Für *wr* und ^*wr* gilt obiges analog für den Schreibzugriff auf Verzeichnisse.

Der Befehl *default* gibt das Verzeichnis an, in dem sich der einloggende Benutzer als erstes befindet. Dieses Verzeichnis muß innerhalb der durch den *rd:*-Parameter gekennzeichneten Verzeichnisstruktur liegen. Eine ungültige Angabe für dieses Verzeichnis quittiert der OS/2-ftpd nicht. Wird unter diesem Benutzernamen ein Verbindungsaufbau vorgenommen, ist der Benutzer nicht in der Lage, den Inhalt dieses Verzeichnisses zu sehen, er ist aber in der Lage, in das korrekte Verzeichnis zu wechseln, wenn er den Namen dieses Verzeichnisses kennt.

```
loginfo: logall
```

Ebenfalls wie der Befehl *default* neu hinzugekommen zum ftpd-Prozess in Warp V4 ist der Befehl *loginfo*. Dieses ermöglicht die Protokollierung der Verbindungsverläufe der ftp-Verbindungen, wenn der *OS/2-SyslogDaemon* (»syslogd.exe«) gestartet wurde. Die Protokollierung erfolgt vom Dateitransfer bis zum Anlegen von Verzeichnissen. Eine Beschreibung hierzu befindet sich im Ordner *Unterstützung* unter *Information|ReadMe|TCP/IP ReadMe-Datei* unter der Rubrik *Neue ftpd-Schlüsselwörter*.

Mit dem OS/2-ftpd ist es möglich, einen kompletten ftp-Server im Internet zu betreiben. Er ist in der Lage, mehrere Verbindungen zuverlässig zu unterhalten und so durch die Stabilität von OS/2 Warp einen exzellenten Internet-ftp-Server abzugeben.

Die Datei »ftpdc.exe«

Im »bin«-Verzeichnis des »tcpip«-Verzeichnisses befinden sich zwei Dateien, die für den ftp-Serverprozeß bei OS/2 verantwortlich sind. Zum einen ist dieses die Datei »ftpd.exe«, die den eigentlichen ftp-Serverprozess darstellt. Die Datei »ftpdc.exe« wird von »ftpd.exe« für jede neu eingehende Verbindung gestartet. Wenn auf einen ftp-Server viele Benutzer gleichzeitig zugreifen wollen, ist sicherlich verständlich, daß man das Computersystem mit entsprechenden Ressourcen (hier Hauptspeicher) ausstatten sollte, um eine akzeptable Verarbeitungsgeschwindigkeit zu erzielen (Stichwort *Paging*).

Triviales ftp. Der tftp-Server unter OS/2

Unter Warp V4 ist es ebenfalls möglich, einen ftp-Server für»tftp« (»triviales ftp«) zu betreiben. Anders als bei ftp erfolgt hier die Authentifizierung der Benutzer ausschließlich über den Rechnernamen des Benutzers. Dieser muß über ein funktionierendes Reversemapping (die Zuordnung von einem Computersystemnamen zu einer IP-Adresse) verfügen, damit dieses funktionieren kann.

Die Datei »tftpauth«

Die Datei »tftpauth« ist eine ASCII-Datei, die mit jedem beliebigen ASCII-Text-Editor editiert werden kann. Die Datei hat einen sehr einfachen Aufbau. Mit dem TCPIP-Konfigurationsprogramm kann auf Seite 3 des Punktes *Sicherheit* ebenfalls diese Datei bearbeitet werden und somit Benutzern Zugriff auf das lokale Computersystem gewährt werden.

```
f:\pub\incoming RW das.system.dort
```

Diese Zeile sagt aus, daß Benutzer des Rechners »das.system.dort« auf Dateien des Verzeichnisses »f:\pub\incoming« zugreifen können. Sie sind sowohl schreib- als auch leseberechtigt. Der prinzipielle Aufbau gliedert sich durch die Angabe eines Verzeichnispfads, der Angabe *RO* für »ReadOnly« (Nur Lesen) oder *RW* für »ReadWrite« (Lesen und Schreiben) und der Bezeichnung eines Computersystems. Es ist möglich, mehrere Computersysteme hintereinander in diese Zeile zu schreiben.

Für jedes andere Verzeichnis wird eine eigene Zeile in dieser Datei benötigt. Erscheint ein Computersystem mehrmals in dieser Konfigurationsdatei, so wird immer der erste Eintrag ausgewertet.

Zusammenfassend ist zu sagen, daß Computersysteme, die über einen Eintrag in »tftpauth« verfügen, dort entsprechend zugeordnet werden, allen anderen Systeme wird der Zugang verwehrt. In solch einem Fall erscheint eine entsprechende Warnung im Serverfenster, daß ein nicht authorisierter Zugriffsversuch stattgefunden hat. Existiert jedoch keine »tftpauth«-Datei, kann jedes Computersystem auf das Standardverzeichnis zugreifen, nicht jedoch auf die Unterverzeichnisse davon. Existiert also die Datei »tftpauth«, so handelt es sich unter OS/2 Warp um ein sehr sicheres Verfahren, um Dateien über tftp zu transferieren.

Das Starten des tftp-Servers

Analog zum Starten von ftpd kann gewählt werden, ob tftpd über den inetd-Superserverprozeß oder separat gestartet werden soll. Wird tftpd über »inetd« gestartet, besteht nicht die Möglichkeit, Parameter zu übergeben.

Wird der tftp-Daemonprozeß gestartet, besteht die Möglichkeit, über den Parameter *-l* die Protokollfunktion zu aktivieren. Auf der Konsole des tftpd wird dann auch angezeigt, was das Standardlaufwerk für tftp-Anfragen ist. Existiert keine »tftpauth«-Konfigurationsdatei im »etc«-Verzeichnis, können nur tftp-Anfragen in diesem Verzeichnis beantwortet werden.

Wird der Parameter *-s* angegeben, ist jedes Computersystem in der Lage, auf diese im Verzeichnis liegenden Dateien zuzugreifen, was in jedem Fall ein beträchtlichtes Sicherheitrisiko darstellt! Mit diesem Parameter wird die Authorisationsfunktion deaktiviert, eine Überprüfung des anfragenden Computersystems unterbleibt.

Wird tftpd ein Verzeichnisname mit vollständigem Pfad übergeben, wird dieses als Standardverzeichnis für tftp-Anfragen benutzt.

ftp made by WorkplaceShell. ftp wie es sein muß

Durch die Objektorientierung von OS/2 mit Objekten für Laufwerke, Dateien und Schablonen für verschiedene Dateiformate bietet es sich förmlich an, ftp objektorientiert zu bedienen. In Warp V4 finden sich mehrere Objekte, die dafür vorgesehen sind und somit für dieses Vorhaben geeignet sind. In früheren Versionen war diese Art von ftp nur durch das Sharewarepaket »Lynn's ftp for WorkplaceShell« möglich, in dieser neuen Version 4 von OS/2 Warp ist eine ähnliche, jedoch nicht ganz so umfangreiche Funktionalität bereits zum Teil integriert.

Die Objekte, die für diese nahtlose Integration in die objektorientierte Arbeitsoberfläche von

OS/2 Warp vorgesehen sind, befinden sich im Ordner *Internetschablonen* im normalen Ordner *Schablonen*. Es handelt sich dabei um die Schablonenobjekte *ftp-Host* und *URL*.

Das URL-Objekt

Über die URL-Schablone kann eine »ftp-URL« definiert werden. Die notwendige Konfiguration dieser URL erfolgt über die Einstellungen dieses so erzeugten Objektes. Wird dieses Objekt durch Doppelklicken geöffnet, wird der voreingestellte WorldWideWeb-Client (in den meisten Fällen der IBM-WebExplorer) gestartet und dieses Dienstprogramm baut eine TCP/IP-Verbindung zu der angegebenen Adresse des ftp-Servers auf, eine richtige Konfiguration vorrausgesetzt. Im Fenster des WWW-Browsers sind die Unterverzeichnisse ersichtlich, und in diesen die Dateien, die dieser ftp-Server anbietet.

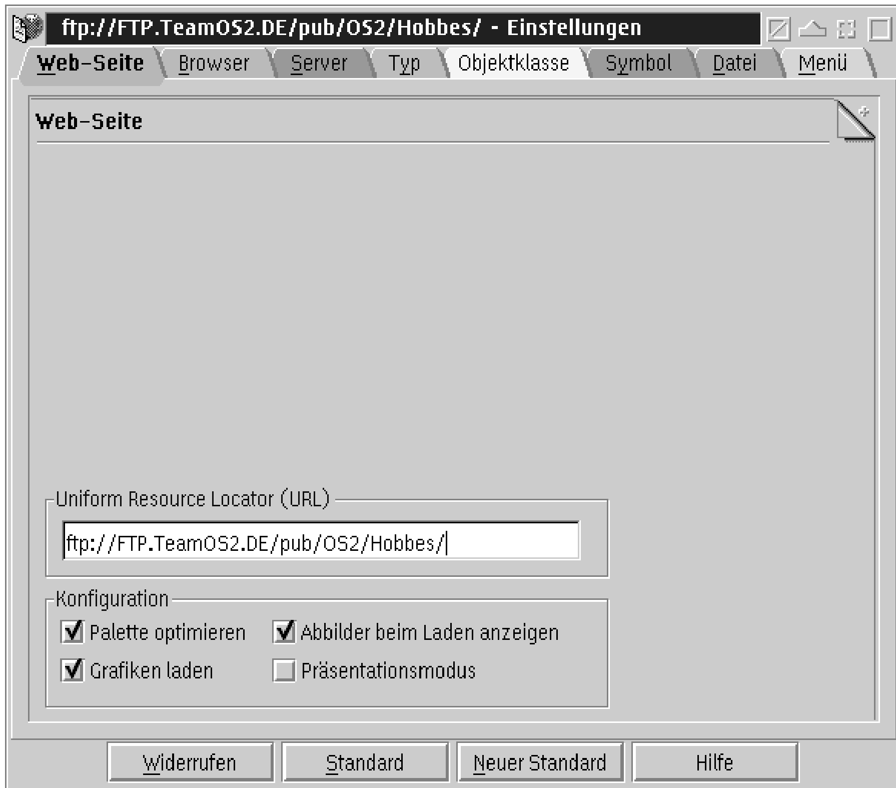


Abb. 9.59: Konfiguration des ftp-URL-Objekts

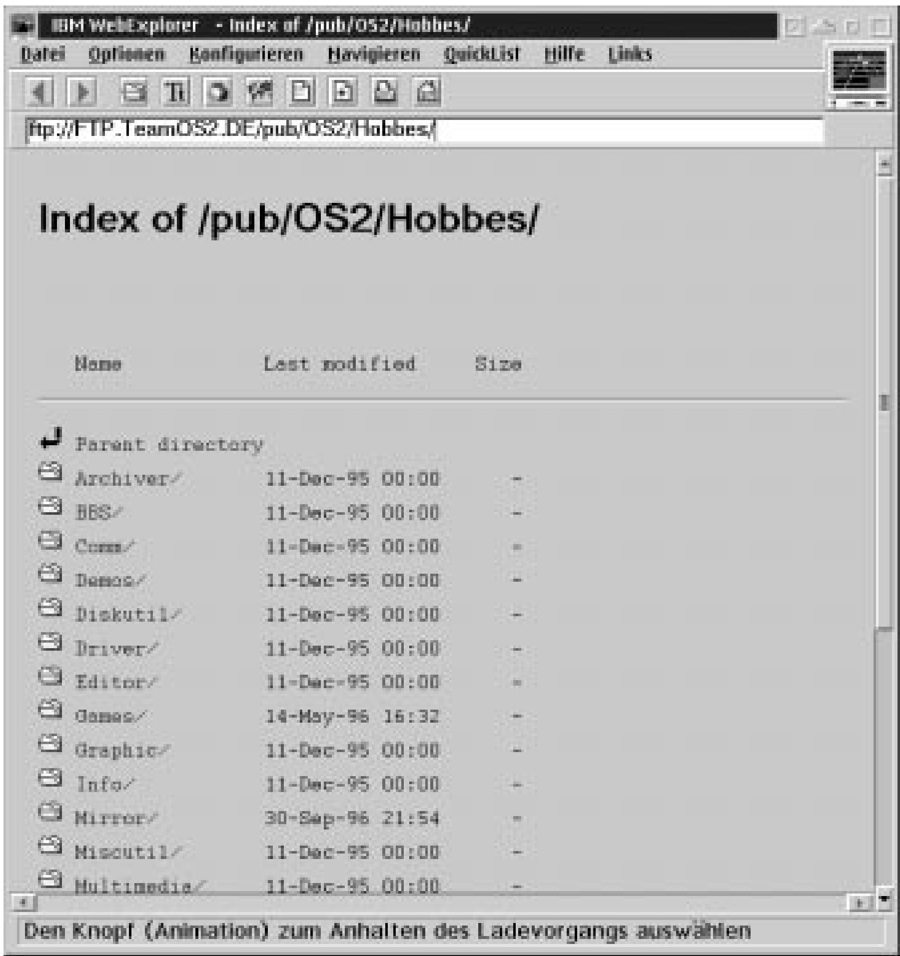


Abb. 9.60: ftp-URL im WebExplorer

Durch Klicken auf die entsprechenden Namen ist es möglich, die Verzeichnisse zu wechseln oder Dateien zu transferieren. In den meisten Fällen wird bei der Übertragung einer Datei eine Meldung ausgegeben, daß für diese Datei keine entsprechende Anzeigefunktion installiert ist und die Datei auf der Festplatte gespeichert werden kann. Durch Bestätigen wird dann der Transfer weitergeführt, die Datei übertragen und auf der Festplatte gespeichert.

Das ftp-Host-Objekt

Bei diesem ftp-Host-Objekt handelt es sich um ein Objekt, das die Adresse und die Zugangsdaten zu einem ftp-Server enthält. Auf der ersten Seite der Einstellungen wird unter den Punk-

ten *Host* die komplette Adresse des ftp-Servers, der gewünschte Benutzernamen und das erforderliche Kennwort eingetragen. Auf Seite 2 ist es möglich, das Startlaufwerk des Servers und das Zielverzeichnis auf dem lokalen Computersystem anzugeben.

Wurde das Objekt korrekt konfiguriert, ist es durch Doppelklick möglich, eine Verbindung zu diesem ftp-Server-System aufzubauen. Es erscheint eine Übersicht über die vorhandenen Dateien und Verzeichnisse, wie es auf dem lokalen Computersystem durch Öffnen des Laufwerksordners für ein Laufwerk bekannt ist.

Mit den vorhandenen Dateiobjekten können Operationen wie Drag&Drop auf ein lokales Laufwerk gemacht und somit die Dateien übertragen und kopiert werden. Das Erstellen einer Referenz ist nicht möglich, das Kopieren von Verzeichnisordnern ebenfalls nicht. Durch Doppelklick auf eine Datei wird diese in das in den Einstellungen angegebene Laufwerk übertragen und geöffnet. Bei Textdateien erscheint der Systemeditor mit dem Inhalt der Textdatei.

Über den Fortschritt der Übertragung wird hier leider nicht informiert, es erscheint lediglich das neue Dateiobjekt auf dem Zielordner, sobald die Übertragung begonnen wurde. Solange die Übertragung noch läuft und nicht beendet ist, kann auch nicht auf das Objekt zugegriffen werden. Es erscheint sofort eine Meldung, daß das Objekt von einem anderen Programm benutzt wird.

Die Ansicht des ftp-Host-Objektes läßt sich auf mehrere Arten öffnen. Durch Doppelklick wird die Standardeinstellung unter *Einstellungen|Menü|Öffnen* zum Anzeigen benutzt. Selektiert man jedoch über das Kontextmenü des Objektes *Öffnen* und wählt die Lasche an, so kann dort neben *Strukturansicht*, *Detailansicht* oder *Symbolansicht* auch eine *Telnetverbindung* oder ein Verbindungsaufbau mit dem auf dem Textmodus basierendem ftp-Programm ausgewählt werden. Bei letzterem muß dann jedoch die komplette Benutzerabfrage von Hand gesteuert werden (sofern keine »NetRC«-Konfigurationsdatei vorhanden ist) und auch die Übertragung der Dateien erfolgt wie oben bereits beschrieben manuell gesteuert.

Wird das Objekt über die Strukturanzeige geöffnet, gilt dieses nur für dieses aktuelle Verzeichnis. Andere Unterverzeichnisse müssen wieder über das selbe Verfahren geöffnet werden. Daher ist es leider nicht möglich (was auch verständlich ist), eine wirkliche Baumstrukturanzeige zu erhalten. In der Detailansicht kann man erahnen, wie groß die Dateien sind, jedoch ist dieses nicht besonders optimal dargestellt.

Das zum Anfang angesprochene Sharewareprodukt »Lynn's ftp for Workplaceshell« (das unter anderem per ftp vom ftp-Server des »TeamOS/2 Deutschland« unter der Adresse »ftp.TeamOS2.de« im Verzeichnis »pub/OS2/Hobbes/Network/TCPIP/ftp/Clients« als Datei »lwpftp11.zip« zu beziehen ist) ist wesentlich ausgeprägter in der Funktionalität. Dort wird angezeigt, wie lange der Transfer dauert, wie groß die Dateien sind, welche Anweisungen noch ausgeführt werden und einiges anderes. Das in Warp V4 integrierte Pendant ist nützlich, aber verglichen mit der Version von Lynn nicht sehr umfangreich.

9.10.6 Computersysteme fernsteuern leicht gemacht

Unter Warp V4 werden Dienstprogramme mitgeliefert, die den Benutzer in die Lage versetzten, einen entfernten Rechner fern zu bedienen oder aus der Ferne das OS/2 Computersystem zu steuern.

Die Dienst- und Hilfsprogramme, die hierfür verwendet werden, befinden sich allesamt im »bin«-Verzeichnis des »tcpip«-Verzeichnisses:

```
telnet.exe
telnetpm.exe und TelnetObjekte der Workplaceshell.
setterm.exe
telnetd.exe
tnlogin.exe und loginunx.exe
```

Wie auch bei den Dienstprogrammen für den Dienst ftp existieren in dieser Version 4 von OS/2 Warp sowohl Dienstprogramme für das Einrichten eines Serverdienstes als auch Programme, um den Telnet-Dienst selber zu nutzen.

Kommandozeilenorientiertes Fernsteuern eines Computers über Telnet

»telnet.exe« ermöglicht es, eine Verbindung zu einem anderen Computersystem über ein Netzwerk aufzubauen. Auf dem Zielcomputersystem muß ein entsprechender Serverprozess (Telnetserver) gestartet sein, damit eine Verbindungsaufnahme erfolgen kann. In der Regel werden Verbindungsaufnahmen durch Benutzerkennung und Passwörter authentifiziert. Durch die Eingabe der richtigen Kombination gewährt der Telnet-Server dann eine zeichenorientierte Kommandoshell (unter Unix bash, tcsh, sh oder andere; unter OS/2 eine normale OS/2-cmd-Shell). Analog zum ftp-Serverprozeß wartet der Telnet-Server auf TCP-Port 23 auf Anfragen.

Mit Telnet ist es auch möglich, andere Internetdienste zu nutzen, wenn auch nicht sehr effektiv. Es ist zum Beispiel denkbar, daß mit Telnet direkt auf den SMTP-Port (Port 25) eines Computersystemes zugegriffen wird, um eine Email direkt über SMTP-Server-Kommandos abzusetzen. An dieser Stelle sei dieses allerdings nicht näher erläutert.

Im etwas veralteten TCP/IP 2.0, das nur als Zusatzpaket zu OS/2 zu erhalten war, war Telnet in der Lage, sich automatisch auf bestimmte Computersystemkonfigurationen bei einer Verbindungsaufnahme einzustellen. Zum Beispiel wenn auf ein SUN-Computersystem zugegriffen werden sollte, wurde die Zeilenhöhe auf 24 Zeilen korrigiert. Dieses ist bei den Version ab Version 3 des TCP/IP-Pakets (seit OS/2 Warp Version 3) nicht mehr der Fall. Teilweise, eine entsprechende Konfiguration der Objekte vorausgesetzt, ist dieses natürlich auch nicht mehr nötig.

Wie auch ftp, kennt Telnet eine ganze Reihe von Kommandozeilenparametern, mit denen entsprechende Konfigurationen erreicht werden können. Um auf obiges Beispiel zurückzukommen, kann mit der Angabe *-h 24* ebenfalls auf 24 Zeilen umgestellt werden. Mit *-w* kann die Spaltenbreite definiert werden. Dabei muß aber beachtet werden, daß die Terminalemulation und der Telnet-Server dieses verarbeiten kann. Zum Beispiel macht es keinen Sinn, eine Spaltenbreite von 200 anzugegeben, da dieses in der Praxis nicht oder nur in den allerseltensten Fällen funktioniert.

Es gibt verschiedene Telnet-Emulationen, die vom OS/2-Telnet-Dienstprogramm emuliert werden können, dennoch ist die Vielfalt der vorhandenen möglichen Telnet-Emulationen viel größer. Zum Beispiel gibt es »VT100«, »VT220«, »ANSI«, »XTERM«, »DECTERM« und noch einige mehr. Telnet kann mitgeteilt werden, welche dieser Emulationen beim Start aktiviert werden soll. Dieses kann natürlich auch nur erfolgen, wenn bekannt ist, mit welcher Emulation der entsprechende Server arbeitet. Der Telnet-Client von Warp V4 versteht zur Zeit nur 100, 220, ANSI, HFT und NVT als Emulationen. Bei anderen Emulationen auf der Serverseite kann es daher unter Umständen zu Zeichenproblemen kommen.

Die meisten Unix-Server bieten nach dem Einloggen auf dem System die Möglichkeit, die Terminalemulation festzulegen. Dieses geschieht entweder über »setenv TERM <Emulationsname>« oder über »set TERM <Emulationsname>«, wobei »TERM« großgeschrieben sein muß, da es sonst auf den meisten Computersystemen nicht als die benötigte Umgebungsvariable erkannt wird. Natürlich sollte hier für *<Emulationsname>* auch nur einer der fünf oben genannten benutzt werden.

Über den Parameter *-e* wird beim Einloggen auf das entfernte Computersystem eine Anzahl an Umgebungsvariablen übergeben, die hier definiert werden. So die Theorie. Leider ist es mir nicht gelungen, dieses auch in der Praxis zu testen, da es bei mir weder mit meiner SUN-SparcStation noch mit einem Telnet auf das OS/2-Computersystem funktioniert hat.

Die Reihenfolge der Parameter sollte einbehalten werden, da zum Beispiel ein »telnet mail.whitehouse.gov -p 25« nicht etwa auf den SMTP-Port des Mailservers des Weißen Hauses geht, sondern nur die Parameterübersicht ausgibt. »telnet -p 25 ...« hingegen liefert das gewünschte Ergebnis. Also den Hostnamen, auf den ein Loginversuch gestartet werden soll, immer am Ende des Befehlsstrings angeben.

Mit dem Parameter *-f* und der Angabe einer Konfigurationsdatei kann die Telnetsitzung konfiguriert werden. Konfigurationsdateien werden mit *Setterm* oder mit dem Objekt *Telnet anpassen* erstellt. Mehr dazu weiter unten im Abschnitt »Setterm.exe, Telnet konfigurieren«.

Um den Fortgang von Telnetsitzungen festzuhalten und aufzuzeichnen, können die Parameter *-l* und *-d* benutzt werden. Dem Parameter *-l* wird außerdem ein Dateiname mit komplettem Pfad übergeben. In diese Datei werden die Daten für das Aufzeichnen der Ein- und Ausgaben geschrieben. Bei *-d* wird ebenfalls eine Datei angegeben, in welche Datei die Debug-Informationen gesichert werden. Das Logfile enthält sämtliche Anweisungen, Login-Prozedurverläufe (jedoch nicht das Passwort beim einloggen, egal, wie oft ein Loginversuch auf andere Computersysteme unternommen wird, wird das Passwort nie mitgeloggt!), Ein- und Ausgaben auf dem Telnetfenster, die im Verlauf dieser Sitzung erfolgt sind. Anders als bei dem Logfile sind im Debug-File sämtliche Angaben, inklusive der Loginpasswörter und Steuerzeichen, vorhanden. Hier ist auch deutlich zu sehen, was gesendet und was im Gegenzug vom Server empfangen wird. Bei der Angabe der Debug-Datei muß also drauf geachtet werden, daß Dritten kein Zugang zu dieser Datei entsteht, sonst kann das Ausspionieren der Passwörter erfolgen. Es ist daher ratsam, auf die Option *-d* ganz zu verzichten, sofern diese nicht dringend (zum Beispiel zur Fehlersuche) benötigt wird.

Telnet via PM, Objekte der Workplace Shell

Im *Internetschablonen*-Ordner gibt es Schablonen für Telnet. Je nachdem, was für ein Netzwerkzugang vorhanden ist, kann einfach per Drag&Drop eine dieser Schablonen genommen und als neues Telnet-Objekt angelegt werden.

In den *Telnet-Einstellungen* kann nun definiert werden, was für ein *Datenstationstyp* (Terminalemulation), was für ein *Tastaturlayout* und was für eine *Zeichenumsetzungstabelle* (Codepageumwandlung) benutzt werden soll. Ferner können hier ebenfalls die Angabe zu der *Protokolldatei* (Debugdatei) und der *Ablaufverfolgungsdatei* (Logfile) angegeben werden. Da es vorkommen kann, daß der TelnetServer nicht auf TCP Port 23, sondern auf einen alternativen Port konfiguriert ist, kann dieser ebenfalls an dieser Stelle spezifiziert werden.

Auf der *Sitzungs-Seite* der Einstellungen kann festgelegt werden, ob es sich um eine Telnetsitzung im Textmodus (Fenster oder Gesamtbildschirm) oder um eine über den Präsentations-Manager ablaufende Sitzung handelt. Für jede Sitzung kann definiert werden, welche Zeilen- und Spaltenauflösung das Fenster annehmen soll. Diese Einstellungen werden dann als Parameter an die Programme weitergereicht.

Die Parameterauswahl des TelnetPM ist gleich der des Textmodus-basierenden Telnets.

»Setterm.exe«, Telnet konfigurieren

Um eine Telnet-Konfigurationsdatei zu erhalten, wird das Programm »setterm.exe« oder das sich im TCP/IP-Ordner unter *Telnet anpassen* befindliche Objekt benutzt. Dort können unter anderem verschiedene Tastaturlayouts erstellt und Farben für die einzelnen Telnet-Emulationen definiert werden. Diese Konfigurationsdateien können auch mit dem grafischen Telnet-Client (»TelnetPM«) benutzt werden, indem dem Parameter *-f* der Konfigurationsdateiname bei einem Aufruf von TelnetPM übergeben wird.

Der Telnetserver in OS/2

Unter Warp V4 ist es möglich, einen Telnet-Dämon-Serverprozeß zu konfigurieren und zu betreiben. Dadurch wird es Benutzern ermöglicht, eine Verbindung zu diesem OS/2-Computersystem über ein TCP/IP-basierendes Netzwerk aufzubauen und dieses OS/2-Computersystem fernzubedienen. Diese Bedienung ist jedoch auf den Textmodus beschränkt.

Wenn der Benutzer per Telnet auf dem OS/2-Rechner eingeloggt ist, kann er, wie für OS/2 üblich, mehrere Prozesse gleichzeitig ausführen lassen. Dabei ist es unwichtig, ob diese Applikationen oder Programme im Textmodus oder in der DOS-Emulation ablaufen oder ob es sich dabei um PM-Programme handelt. Es ist in jedem Fall lediglich nur möglich, auf in dieser Telnet-Kommando-Shell ablaufende Programme zu reagieren, wobei dies bei Programmen unter der DOS-Emulation nicht der Fall ist. Eingaben und Ausgaben an Programme, die durch die Befehle *start* oder *detach* entweder im Textmodus oder auf der Arbeitsoberfläche gestartet wurden, können ebenfalls nicht getätigt werden. Außerdem ist es in keinem Fall ratsam, einen Wechsel zu einem dieser Programme zu vollführen, da dann keine Möglichkeit besteht, wieder Kontrolle über diese Telnet-Kommando-Shell zu erreichen. Die Sitzung scheint dann tot oder abgestürzt zu sein.

Wurde ein PM-Programm gestartet und dieses nicht per *start* als separater Prozeß, so ist diese Telnet-Sitzung nicht weiter nutzbar (wie oben beschrieben), jedoch kann die Kontrolle über diese Telnetsitzung wiedererlangt werden, wenn das gestartete PM-Programm beendet wird. Existiert ein Textmodus-Utility zur Prozeßverwaltung (Beispiel »go«, »pstat« und »killem«), so kann der Benutzer über eine weitere Telnetverbindung diesen PM-Prozeß abschießen und somit beenden. Dann gewinnt er auch wieder Kontrolle über diese erste Telnetsitzung.

Die Konfiguration des Zugangspasswortes für den Telnetzugang zu dem lokalen OS/2-Rechner wird über das TCP/IP-Konfigurationsobjekt getätigt. Dabei handelt es sich um einen Eintrag auf der Seite *Sicherheit*. Das Passwort wird als Umgebungsvariable »telnet.password.id« in die Datei »config.sys« eingetragen und ist dann nach einem Systemneustart aktiv. Das Passwort ist in der »config.sys« klar und unverschlüsselt lesbar und so unter Umständen für jeden frei zugänglich. Alternativen mit verschlüsseltem bzw. gekryptetem Passwort werden weiter unten besprochen. Ist kein Passwort definiert, so ist auch kein Einloggen über Telnet möglich.

Jetzt ist ein Benutzer über »telnet os2.computer.system« in der Lage, eine Telnetverbindung mit diesem System aufzubauen. Ist er eingeloggt und hat eine OS/2-Kommando-Shell, so ist dieser Benutzer in der Lage, sämtliche OS/2-Befehle, die ein normaler Benutzer durch Aufrufen einer OS/2 Befehlszeile tätigen könnte, ebenfalls zu unternehmen. Dazu gehören auch die Programme »FDisk« und »Format«!

Wie auch beim ftp-Server ist es möglich, dem Telnet-Daemon über Kommandozeilenparameter andere Einstellungen mitzuteilen. Durch *-u* ist es möglich, ein anderes als das in der »config.sys« eingetragene Passwort zu benutzen.

Wird TelnetD mit dem Parameter *-p* und einer Portnummer gestartet, so erwartet der Telnet-Dämon Anfragen auf diesem alternativen Port und nicht auf dem *Standardport TCP 23*.

Kombiniert mit dem Parameter für das Passwort kann so ein recht nützlicher Schutz erlangt werden. Selbst wenn eine nicht berechtigte Person in der Lage war, die »config.sys« anzuschauen und so über das scheinbar verwendete Passwort Kenntnis zu erlangen. Das Passwort, das dort definiert wird, wird dann ja nicht benutzt und ein normaler Telnet bekommt keine Verbindung, weil auf einem falschen Zielport eine Verbindungsaufnahme versucht wird.

Um kontrollieren zu können, was genau durch eine Telnetsitzung geschieht, besteht die Möglichkeit, das Logging oder auch die Protokollierung einzuschalten. Dieses geschieht durch den Parameter *-d* (Debugmodus aktivieren) und der Angabe *-l*, die eine Datei »telnetd.log« im Root-Verzeichnis des Bootlaufwerkes anlegt. In dieser Datei stehen sämtliche Zeichen, die empfangen und gesendet werden, selbst die Passwörter sind hieraus ersichtlich. Jede Eingabe wird aufgezeichnet und kann später ausgewertet werden, wobei dieses jedoch recht umständlich ist und die Logfiles sehr schnell recht groß werden können.

Hat ein fehlerhaftes Programm einen Fehler auf dem System (z.B. einen Sys 3175), auf dem der Telnet-Daemon in Betrieb ist, verursacht, so ist es nicht möglich, sich auf dieses OS/2-Computersystem einzuloggen, solange das Fenster mit der Fehlermeldung vorhanden ist. Dieses Fenster muß erst durch Klicken auf das Feld *Beenden* beendet werden, damit wieder ein Prozeß gestartet werden kann. Dieses liegt jedoch nicht am OS/2-Telnet-Daemon, sondern vielmehr an der Prozeßverwaltung von OS/2.

Einen großen Nachteil hat diese »Out-of-the-box«-Konfiguration des Telnet-Daemons unter Warp V4 jedoch. Es wird beim Verbindungsaufbau nur nach dem Passwort gefragt, aber nicht nach einem Benutzernamen.

Telnetloginprozeduren unter OS/2 Warp

Sobald ein Verbindungsaufbau erfolgt ist, startet der Telnet-Daemon das Programm »telnetdc.exe«, das dann seinerseits »tnlogin.exe« startet, um die Loginprozedur abzuarbeiten.

Das Programm »tnlogin.exe« fragt bei der Loginprozedur nur nach einem Passwort, nicht jedoch nach einem Benutzernamen. Das verwendete Passwort befindet sich im Klartext in der »config.sys«. Wenn der Aufruf des Telnet-Daemons mit dem Parameter *-u* und einem anderen Passwort erfolgt ist, wird dieses stattdessen benutzt.

Es gibt unter Warp V4 aber auch die Möglichkeit, eine Loginprozedur zu konfigurieren, die zuerst nach einem Benutzernamen und dann nach einem Kennwort fragt. Das Programm »loginunx.exe« (LoginUnix) benötigt dazu die Datei »passwd« im »etc«-Verzeichnis (Umgebungsvariable!). Diese Datei erinnert ganz stark an die Password-Datei von Unix-Systemen und sieht auch genauso aus. Dort stehen die Benutzerkennwörter nicht mehr direkt lesbar, sondern verschlüsselt. Dadurch wird eine gewisse Sicherheit erreicht.

Um dieses Programm zur Loginprozedur zu benutzen, sind zwei OS/2-Kommandozeilenbefehle nötig. Diese werden beide im »bin«-Verzeichnis des »tcpip«-Verzeichnisses ausgeführt:

```
ren tnlogin.exe tnlogin.ex_
ren loginunx.exe tnlogin.exe
```

Dadurch wird erwirkt, daß ab sofort für die Telnet-Loginprozedur das loginunx-Programm verwendet wird und nach einem Benutzernamen und dann nach einem Kennwort gefragt wird. Um dieses jedoch zu realisieren, sollte vorher in jedem Fall eine gültige »passwd« erzeugt werden. Ist diese nicht vorhanden, ist es nicht mehr möglich, eine Verbindung mit diesem OS/2-Rechner über Telnet aufzubauen. Leider existiert jedoch keine Beispiel-Passwd. In diesem Fall ist es die leichteste Möglichkeit, eine Passwd von einem Unix-System zu besorgen und diese dann zu verwenden.

Ein Dienstprogramm zum Bearbeiten einer existierenden »passwd« ist ebenfalls bei OS/2 nicht standardmäßig vorhanden, aber es gibt auf dem ftp-Server »Hobbes.NMSU.Edu« mit der URL »ftp://hobbes.nmsu.edu/os2/network/tcpip/passwd.zip« ein Archiv, in dem zwei Programme zum Bearbeiten dieser »passwd« vorhanden sind. Wie sie funktionieren, ist der Anleitung des Pakets zu entnehmen.

Wird während der Login-Phase eine ungültige Kombination aus Benutzername und Passwort eingegeben, wird diese Verbindung vom OS/2-Telnet-Daemon abgebrochen und beendet. Selbst ein Vertippen verursacht dieses Vorgehen. Wird die Standardinstallationsvariante des Login-Programms verwendet, ist es möglich, beliebig viele Fehlversuche bei der Passworteingabe zu tätigen. In keinem Fall wird die Verbindung beendet.

9.10.7 Anweisungen auf einem entfernten Computersystem ausführen

Es gibt zwei grundlegende Möglichkeiten, Programme und Befehle auf einem entfernten Computersystem auszuführen, ohne sich auf dem eigentlichen Computersystem einloggen zu müssen. Zum einen ist dieses durch die »RemoteShell RSH« möglich, zum anderen durch »RemoteExec REXEC«. Der Unterschied zwischen beiden hinsichtlich der Ausführung der Anweisungen ist nicht sehr groß, bei RSH erfolgt die Überprüfung der Autorisierung über den Computersystemnamen oder die IP-Nummer, bei REXEC erfolgt dieses über einen Benutzernamen und das zugehörige Password.

Die Funktionsweise dieser Dienste, sowohl als Client als auch die Servervariante unter Warp V4, wird im folgenden näher erläutert.

Vorweg ein paar allgemeine Anmerkungen zu den Remote-Access-Serverdiensten von OS/2.

Es ist einem auf das OS/2-Computersystem per »RShell« oder »RExec« zugreifendem Benutzer generell möglich, sämtliche Programme aufzurufen, die auf dem OS/2-Rechner existieren, ganz abgesehen davon, ob er in der Lage ist, diese dann auch noch zu benutzen oder zu bedienen. Ganz besonders sollte Vorsicht bei den Programmen »FDisk« oder »Format« walten, da dadurch verheerende Datenverluste im Mißbrauchsfall kommen könnten

Jeder OS/2-Textmodus-Shell-Befehl ist ausführbar, jedoch erfolgt keinerlei Zeichenemulation für diese Sitzung, da es sich ja nicht um einen auf solche Vorgehensweise ausgelegten Dienst handelt, und daher entstehen unter Umständen Probleme bei der Zeichenumsetzung.

Generell ist es auch nur möglich, mit OS/2-Programmen im Textmodus zu arbeiten oder diese sinnvoll zu starten und zu steuern. Handelt es sich bei den anderen Programmen um selbständig terminierende Programme (egal ob DOS, PM oder OS/2), so ist die Ausführung natürlich auch möglich. Erfordert das gestartete Programm jedoch eine Interaktion mit dem Benutzer, kann dieses Programm im Fall für DOS-, Windows- oder PM-Programme nicht sinnvoll ausgeführt werden, da eben eine solche Interaktion nicht möglich ist.

Folgendes Problem entsteht daraus: Der Serverprozeß führt dieses Programm aus und ist dann nicht mehr als Serverprozeß verfügbar, bis das gestartete Programm beendet wird. Da ein Benutzer aber nur in der Lage ist, Textmodus-OS/2 Programme zu steuern (weil sonst ein weiterer Prozeß gestartet wird und auf diesen separaten Prozeß nicht von der RSH- oder RExec-Shell zugegriffen werden kann), können solche Programme auch nicht beendet werden. Folglich bleibt der Serverprozeß immer in diesem Stadium, bis ein Timeout auftritt. Im schlimmsten Fall jedoch für immer, wenn der Timeout deaktiviert wurde.

Erfolgt ein Zugriff per RemoteShell oder -Exec auf den OS/2-Rechner, so kann zu diesem Zeitpunkt kein weiterer Zugriffsversuch abgearbeitet werden. Wenn der erste Zugriff beendet wurde, wird der zweite direkt danach ausgeführt.

RemoteShell, die entfernte Shell zum Arbeiten

Über »RemoteShell« können auf entfernten Computersystemen Anweisungen ausgeführt werden, ohne daß ein Login über Telnet oder einen vergleichbaren Dienst zu erfolgen brauchen.

RSH, der RemoteShell-Client

Mit dem Dienstprogramm »rsh.exe« (Remote Shell) ist ein Benutzer in der Lage, ein Shell-Befehl auf einem entfernten Computersystem auszuführen. Dabei muß auf dem entsprechenden System ein RSH-Daemon (wartet auf Port TCP 514 auf Anfragen) installiert und in Betrieb sein.

Durch Aufruf von »rsh« ohne Parameter wird eine kurze Erläuterung ausgegeben, wie der Aufruf zur erfolgen hat. Der Aufbau ist sehr einfach. »rsh computer.system.name -l Benutzeridentifikation <auszuführendes Shellkommando>« führt den entsprechenden Befehl auf diesem Computersystem aus. Dabei ist zu beachten, daß in der Regel eine besondere Vorkehrung auf dem Server zu treffen ist. Es muß eine Zugangsberechtigung für den Zugang mit »rsh« angelegt werden.

Ist die Umgebungsvariable *User* gesetzt, so kann diese für den Parameter *-l* benutzt werden, das heißt, der Parameter *-l* wird weggelassen und der RSH-Client ergänzt automatisch diesen Parameter durch den Wert, der durch die Umgebungsvariable gesetzt wird.

Wie das unter OS/2 funktioniert, werden wir weiter unten sehen. Auf Computersystemen, die unter dem Betriebssystem Unix arbeiten, wird in der Regel diese Zugangsberechtigung über eine Datei im Home-Verzeichnis des Benutzers geschaffen. Diese Datei heißt ».rhosts« und enthält den Computersystemnamen oder die IP-Nummer, von dem der Zugriff mit RSH erfolgen soll.

Dieses wirft natürlich eine nicht unwesentliche Sicherheitslücke auf, denn jeder Benutzer eines Systems, auf den diese Zugriffsberechtigung zutrifft, ist in der Lage, auf diesen Server mit RSH zuzugreifen, wenn er den entsprechenden Benutzernamen kennt (was nicht sonderlich schwer ist!).

Daher sollte RSH in jedem Fall mit Vorsicht zu genießen sein und nach Möglichkeit nicht verwendet werden. Effektivere Methoden zur Fernsteuerung sind über das Dienstprogramm »Telnet« möglich.

Es ist unter der Benutzung von RSH auch möglich, Dateitransfers zu tätigen. Dazu muß lediglich bei den auszuführenden Anweisungen eine Umleitung der Eingabe bzw. der Ausgabe erfolgen. Das folgende Beispiel kann das vielleicht ganz gut erklären: »rsh -b host.name »cat < dateiname.remote« > dateiname.lokal«. Hierdurch wird die Datei »dateiname.remote« von einem Unix-System namens »host.name« (oder von einem beliebigen anderen System, auf dem das Unix-Tool »cat.exe« existiert) auf das lokale System unter dem Namen »dateiname.lokal« gespeichert. Der Parameter *-b* teilt RSH mit, daß es sich um eine Binärübertragung handelt. Der umgekehrte Weg ist genauso machbar, dabei muß lediglich die Richtung der Pfeile umgekehrt werden.

Der RemoteShell-Server

Es ist unter Warp V4 möglich, einen RemoteShell-Server (»rshd.exe«) zu betreiben. Nachfolgend ist das Verfahren der Konfiguration und was zu beachten ist, wo Probleme auftauchen können, näher erläutert.

Die wichtigste Voraussetzung für den Betrieb ist das Anlegen der Datei »rhosts« im »etc«-Verzeichnis (Umgebungsvariable beachten!). In dieser Datei werden die Zugangsberechtigungen für den RSH-Dienst definiert. Diese Datei ist eine ASCII-Datei und kann mit jedem beliebigen Editor oder mit Hilfe des TCP/IP-Konfigurationsobjektes bearbeitet werden. Der Aufbau der Datei ist recht simpel. Pro Zeile wird ein Benutzer definiert, der per RSH auf das OS/2-Computersystem per RSH zugreifen darf.

```
9.11.112.9 janet
9.11.113.24 roger
138.169.14.2 mike
workstation-3.ny.us.ibm.com lou
```

Durch diese Angabe wird den Benutzern Janet, Roger, Mike und Lou ein Zugriff auf diesen OS/2-Rechner gestattet. Allerdings können die Zugriffe nur von definierten Rechnern erfolgen, da eine Überprüfung der IP-Adresse stattfindet. Erfolgen die Zugriffe von anderen Systemen, werden die Verbindungen abgebrochen, weil keine entsprechenden Zugriffsberechtigungen vorhanden sind.

Statt der Angabe der IP-Nummer kann auch der Name des Computersystems angegeben werden. Zur Überprüfung wird die sich anmeldende Rechneradresse über Reverse-Mapping zurück in einen Namen gewandelt. Dieser muß dann mit dem angegebenen Namen übereinstimmen, um einen Zugriff zu gewährleisten.

Wird nur ein Computersystemname oder eine IP-Nummer angegeben, nicht aber ein Benutzername, so ist der Zugriff per RSH von jedem Benutzer dort möglich.

Wenn die Datei »rhosts« nachträglich editiert wird, ist ein Neustart des RSH-Daemons notwendig, um die aktualisierten Daten auch benutzen zu können.

Nachdem diese Datei angelegt wurde, kann der RSH-Daemon über die Eingabe von »rshd.exe« gestartet werden. Über das TCP/IP-Konfigurationsobjekt kann auch definiert werden, ob der RSH-Daemon beim Systemstart gestartet und ob dieses über den inetdaemon-Superserverprozeß erfolgen soll. Wird der Daemon über den Inet-Daemon gestartet, kann keine Umleitung der Ausgabe in eine Datei erfolgen.

Der RSH-Daemon wird ohne Parameter gestartet. Es besteht die Möglichkeit, über die Umgebungsvariable »RSHD.TIMEOUT.IN.SECONDS« einen Timeout für RSH-Verbindungen zu spezifizieren. Dieses geschieht über die Angabe »set RSHD.TIMEOUT.IN. SECONDS=<Anzahl der Sekunden>« in der »config.sys« oder vor dem Aufruf des RSH-Daemon-Programms.

Wird eine RSH-Verbindung aufgebaut, ist dies im Fenster des RSH-Daemons zu verfolgen. Dort wird auch die Ausgabe, die Befehle und das Beenden der Verbindung wiedergegeben. Der Rechnername, von dem der RSH-Verbindungsaufbau ausging, wird leider nicht angezeigt.

Jedoch wird der Benutzer angezeigt, der diese Zugriffsberechtigung benutzt. Es wird bei mehreren Benutzern des gleichen Computersystems so lange das »rhosts«-File abgearbeitet, bis eine entsprechende Zuordnung getroffen werden kann. Dann ist es ein leichtes, anhand dieser Datei nachzuvollziehen, wer der Benutzer gewesen ist. Wenn jedoch mehrere Benutzer gleichen Namens von unterschiedlichen Computersystemen eine Zugriffsberechtigung haben, so ist es nicht möglich, definitiv zu sagen, von welchem System der Zugriff erfolgt ist.

Es ist leider nicht über Kommandozeilenparameter möglich, ein Logfile von RSH-Verbindungen anzulegen. Da hilft aber der folgende Trick. Wenn der RSHD über »rshd.exe« gestartet wird, wird einfach als Parameter > *<Dateiname>* angegeben. Der gesamte generierte Output wird so in diese Datei umgelenkt und man hat ein wunderbares Logfile, in dem alles steht, was geschehen ist.

RemoteExec, Ausführen von Programmen auf dem entfernten System

Ähnlich wie »RSH« kann über »RExec« auf einem entfernten Computersystem ebenfalls ein Shell-Kommando oder ein Programm ausgeführt werden. Anders als bei »RSH« erfolgt bei »RExec« eine Authentifizierung über eine gültige Benutzername-Passwortkombination.

Der RExec-Client

Die Ausführung von »rexec.exe« erfolgt prinzipiell auf die gleiche Weise wie bei »rsh.exe«. Diesem Dienstprogramm werden als Parameter ein Zielcomputersystem und eine auszuführende Anweisung übergeben.

In diesem Fall wird von »RExec« eine Rückfrage nach einem Benutzernamen und einem Kennwort für diesen Benutzer getätigt. Erst dann erfolgt der Verbindungsaufbau zum RExec-Server (dieser erwartet Anfragen auf Port TCP 512).

Es besteht auch die Möglichkeit, diese zugangsrelevanten Daten über weitere Kommandozeilenparameter zu spezifizieren. Durch die Angabe *-l* gefolgt von dem Benutzernamen wird dieser an »RExec« übergeben und die Übergabe des Passwortes erfolgt durch Benutzung des Parameters *-p*.

Ist die Datei »NETRC« im »etc«-Verzeichnis vorhanden, so werden die Angaben für Benutzername und Password für diesen, entsprechend in dieser Datei definierten Host benutzt. Es erfolgt dann keine zusätzliche Abfrage, und eine Eingabe über die Kommandozeile ist nicht mehr nötig.

Um zu vermeiden, daß Definitionen aus dieser Datei verwendet werden, ist der Parameter *-n* zu verwenden.

Sollen im Verlauf dieser »RExec«-Anwendung binäre Daten transferiert werden, muß der Aufruf ebenfalls analog zu »RSH« mit dem Parameter *-b* erfolgen. Das Übertragen einer Datei kann analog zu der Verwendung von »RSH« über eine Umleitung der Ein- und Ausgabe erfolgen.

Ist es gewünscht, interaktiv mit dieser Shell zu kommunizieren, ist der Parameter *-i* dafür zu benutzen.

Der RExec-Serverprozess

Damit auf dem OS/2-Rechner ein RemoteExec-Serverprozess (»rexecd.exe«) gestartet werden kann, müssen über die Umgebungsvariablen die Begriffe *user* und *passwd* definiert werden. In aller Regel erfolgt dieses über »set user=name« und »set passwd=password« in der »config.sys« oder in einer OS/2-Kommandoshell. Jedoch ist es bei der Eingabe in einer Kommandoshell auch nur für diese Kommandoshell gültig. Der Aufruf zum Starten des Daemonprozesses muß folglich in dieser Shell stattfinden.

Aufgerufen wird der OS/2 RExec-Server durch *rexecd*. Es werden keine Parameter übergeben. Über das TCP/IP-Konfigurationsobjekt ist ebenfalls eine Konfiguration möglich, wie der Serverprozeß gestartet werden soll.

Der einzige Parameter, der übergeben werden kann, ist die Angabe einer Nummer. Auf diesem versucht der »Rexecd« einen Socket zu belegen, was bei mir jedoch bei jedem Versuch mit einer Fehlermeldung fehl schlug. Über dieses Problem existieren leider keine Online-Unterlagen.

In der »config.sys« kann eine Umgebungsvariable »REXECDTIME« definiert werden, die den Timeout für RExec-Verbindungen festlegt. Dieses erfolgt durch »set REXECDTIME=zeitintervall«. Dieser Parameter sollte aus eingangs erläuterten Ursachen nicht zu hoch angesetzt werden. 30 Sekunden sind sicherlich eine akzeptable Zeitspanne, bis ein über »RExec« ausgeführte Anweisung beendet wird.

9.10.8 Wer ist eigentlich der Benutzer »@Computer.system.name«?

Mit Hilfe von »Finger« (unter Warp V4 lautet der Name des Programms »finger.exe«) ist es unter anderem möglich, festzustellen, welche Benutzer auf einem fernen Computersystem eingeloggt sind. Dazu muß allerdings auf dem Zielsystem ein Fingerdaemon in Betrieb sein, der die Anfragen (auf Port TCP 79) auswertet und so Auskunft über die eingeloggten Benutzer gibt.

Der Programmaufruf ist simpel. Als Parameter wird der Name des zu befragenden Computersystems angegeben. Dadurch wird eine Liste der eingeloggten Benutzer generiert. Wird eine leere Liste zurückgegeben, ist kein Benutzer auf diesem Computersystem eingeloggt.

Durch den Aufruf mit »<User>@computer.system.name« als Parameter werden nähere Informationen über den eingeloggten Benutzer *<User>* ausgegeben, sofern diese vorhanden und auch zugänglich sind.

Diese Informationen können Telefonnummern, Adressen, Email-Adressen, Projektinformationen sowie die Uhrzeit des Logins, wie lange der Benutzer keine Eingabe getätigt hat bis zu der Angabe einer kleinen Nachricht an den das »Finger«-Programm benutzenden Anwender beinhalten. Dieses ist stark davon abhängig, welcher Fingerdaemon installiert wurde und wie er konfiguriert ist.

Es gibt noch wesentlich mehr Ausgabemöglichkeiten als die oben geschilderten. So ist es mit »Finger« auch möglich, auf Datenbanken zuzugreifen, wenn auf dem entsprechenden Server ein Programm in Betrieb ist, das dieses ermöglicht. Beispiele hierzu sind die Datenbanken des »Deutschen Network Information Center« (De-NIC) und vom Internic (oberste Institution bei der Vergabe von Domainnamen).

Mit Warp V4 wird für OS/2 kein Finger-Daemon-Dienstprogramm ausgeliefert. Es besteht jedoch die Möglichkeit, dieses Programm per ftp zu beziehen. Zum Beispiel ist ein auf dem Textmodus basierender Fingerdaemon auf dem ftp-Server des TeamOS/2 Deutschland (»ftp.TeamOS2.de«) im Verzeichnis »pub/OS2/Hobbes/Network/TCPIP/Tools/Finger« mit dem Dateinamen »fingerd2.zip« zu beziehen.

9.10.9 Talk: Reden über die Tastatur? Wie denn das?

Über das TCP/IP-basierende Netzwerk Internet können nicht nur passiv Nachrichten versandt werden (zum Beispiel durch den Dienst »electronic Mail«, kurz »Email«), es kann auch eine personenbezogene direkte Diskussion geführt werden, die zwischen zwei Personen stattfindet. Dieses geschieht über das Dienstprogramm »Talk«, das im Lieferumfang von Warp V4 vorhanden ist. Dazu gehört auch das Programm »talkd.exe«, das einen entsprechenden Serverprozess bereitstellt, damit dieser Dienst etabliert werden kann.

Wie funktioniert der Verbindungsaufbau?

Das Verfahren ist recht einfach zu verstehen. Ein Talk-Client versucht, eine Verbindung zu einem Talk-Daemon (auf Port UDP 517) aufzubauen. Der Talk-Daemon auf dem entfernten System meldet dem Benutzer auf dem entfernten System eine eingehende Talk-Aufforderung und die Adresse, von der diese Anforderung ausgeht. Der Benutzer des angetalkten Systems seinerseits startet ebenfalls einen Talk-Client und richtet diesen auf die Adresse, von der die eigentliche Aufforderung hervorging. Der zweite Talk-Client versucht nun seinerseits, eine Verbindung zum Talk-Daemon des ersten Computersystems aufzubauen (auch wieder auf Port UDP 517). Wenn dieses erfolgt ist, besteht eine aktive Verbindung.

Dieses erklärt auch, warum auf jedem System, auf dem Talk eingesetzt werden soll, ein Talk-Daemon in Betrieb sein muß.

Bei OS/2 Warp wird der Talk-Daemon automatisch gestartet, wenn versucht wird, eine Verbindung mit Talk aufzubauen und der Talk-Daemon vorher noch nicht gestartet gewesen ist.

Unter OS/2 Warp besteht die Möglichkeit, über eine Umgebungsvariable »USER« einen für Talk relevanten Benutzernamen festzulegen (zum Beispiel in der »config.sys«: »set user=<username>«). Das vorhandene Standard-Talk-Dienstprogramm benutzt diese Variable, um Anfragen zu bearbeiten. Leider gibt der Talk-Daemon keinerlei Auskunft darüber, welcher Benutzer von außen angesprochen werden soll, es wird lediglich mitgeteilt, daß ein Verbindungwunsch eingegangen ist und von welcher Ursprungsadresse dieses erfolgt ist.

Wenn ein Benutzer eines OS/2-Rechners angesprochen werden soll, muß die Talk-Anforderung an den Benutzer »*user*@computer.system.name« gerichtet sein, wobei für die Angabe

»computer.system.name« der vollständig domainisierte Domainname (FQDN, fully qualified domain name) eingesetzt wird. *user* entspricht der eventuell gesetzten Umgebungsvariable auf dem OS/2-Zielsystem. Ist diese Variable nicht gesetzt, wird standardmäßig *os2user* als Benutzername verwendet.

Versucht ein Benutzer eines entfernten Systems einen auf diesem OS/2-Rechner nicht vorhandenen Benutzer (also einen anderen Benutzer, als über die *User*-Variable definiert) anzusprechen, kann es geschehen, daß – obwohl dieser an die richtige Adresse zurückspricht – dennoch keine Verbindung aufgebaut wird, weil der TalkDaemon des Gesprächspartners einen anderen Benutzernamen erwartet.

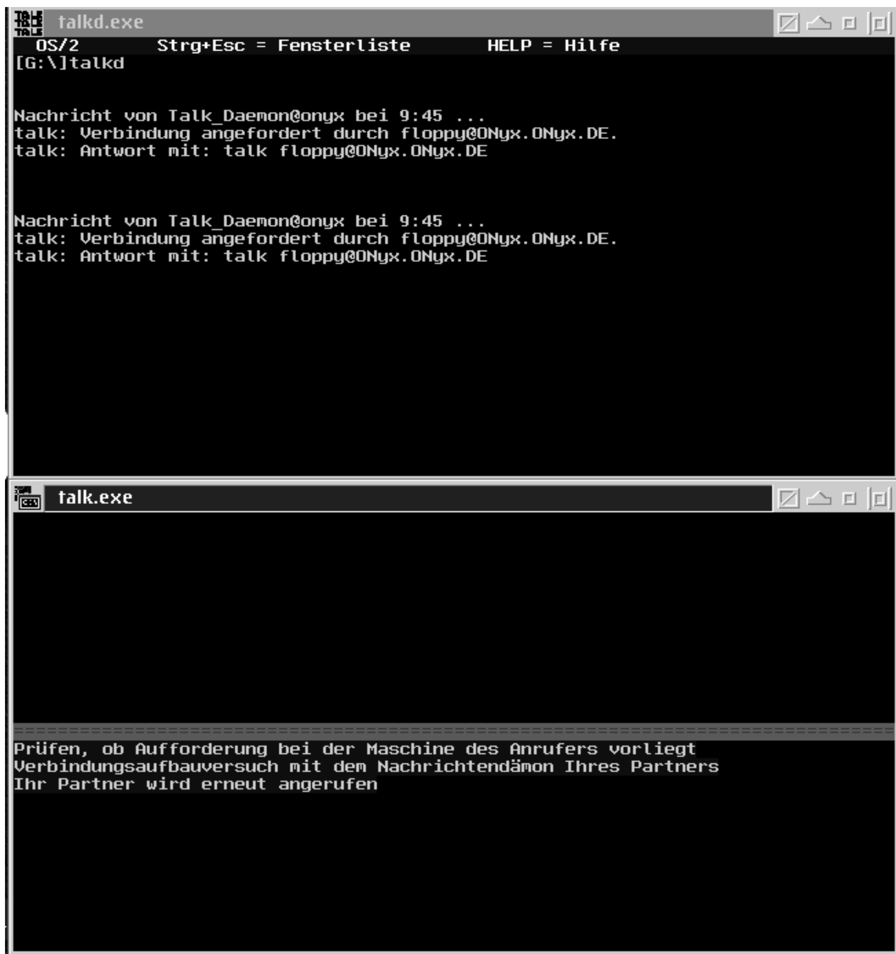


Abb. 9.61: TalkD und Talk im Einsatz

Selbstgespräche?

Es geht auch, mit sich selbst eine Talk-Verbindung aufzubauen. Dazu wird einfach über den Befehl *talk os2user@hostname* eine Anforderung gestartet. Ist der Talk-Daemon noch nicht gestartet, erfolgt das jetzt. Der Talk-Daemon gibt dann eine Anforderung von einem »os2user@hostname« an. Durch einen weiteren Aufruf von *talk os2user@hostname* kann dann eine Verbindung aufgebaut werden und es so ermöglicht werden, miteinander zu talken. Der Begriff »os2user« muß dann jedoch durch den über die Umgebungsvariable definierten Namen ersetzt werden.

Der Hintergedanke bei diesem Vorgehen ist: Ein Benutzer, der durch Telnet auf den OS/2-Rechner gelangt, möchte mit der Person, die vor dem Bildschirm sitzt, eine Kommunikationsverbindung über »Talk« eingehen. Dazu rufen beide Seiten »Talk« auf, wobei der vor dem Bildschirm sitzende Benutzer vorher schon den Talk-Daemon gestartet haben sollte. Dann wird zwischen beiden eine Talk-Verbindung etabliert und die Unterhaltung kann erfolgen.

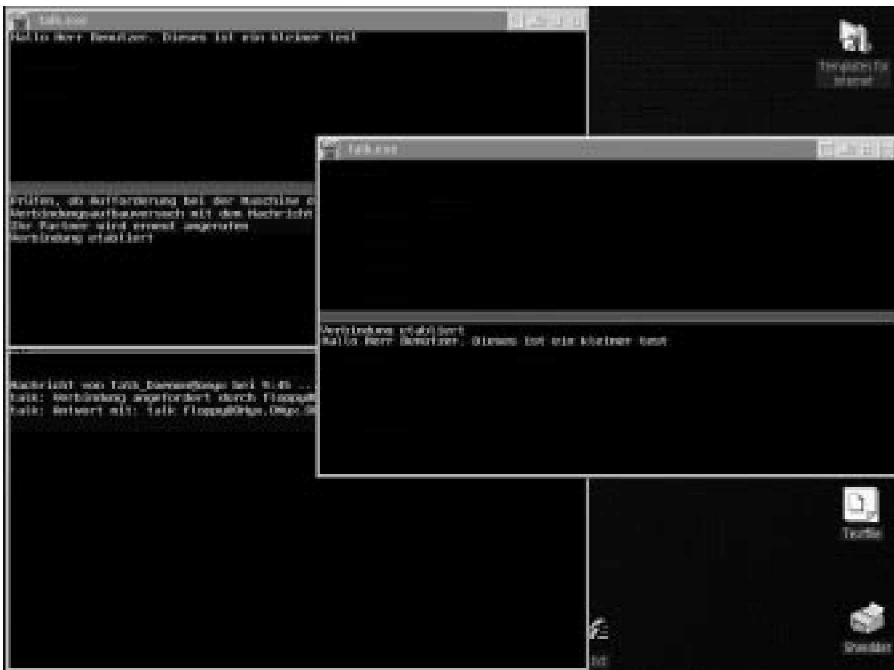


Abb. 9.62: Selbstgespräche mit Talk ...

Wissenswertes über Talk

Die Zeichen, die eingegeben werden, sind bei dem Gesprächspartner erst dann sichtbar, wenn [Enter] betätigt wurde. Auf den meisten Unix-Arbeitsstationen kann dies jedoch unterschied-

lich sein, dort erscheinen in der Regel jedoch die Zeichen auf der Gegenseite schon während des Eingebens.

Durch einen Talk können immer nur exakt zwei Personen miteinander kommunizieren. Es spricht jedoch nichts dagegen, beliebig viele Instanzen des Programms »Talk« zu starten und mit beliebig vielen Personen zu reden. Jedoch wird dieses auf die Dauer schnell unübersichtlich. Ein Multi-Benutzer-Talk ist der Internetdienst IRC (Internet Relay Chat), in dem virtuell mit unbegrenzt vielen Benutzern gleichzeitig und dazu auch noch übersichtlich kommuniziert werden kann. Ein entsprechendes Dienstprogramm liegt dieser Version von OS/2 Warp auf der Bonus-CD bei. Es gibt jedoch verschiedene andere Versionen dieses Dienst-Programms als Shareware. Über den Dienst ftp sind eine Auswahl unter anderem unter folgender Adresse abrufbar: »ftp://ftp.teamos2.de/pub/OS2/Hobbes/Network/TCPIP/IRC/Clients«

NTalk und YTalk

Neuere Versionen von Talk-Programmen unterstützen die Dienste »NTalk« (setzt auf Port UDP 518 auf) oder »YTalk«. »NTalk« wird sehr häufig auf Unix-Systemen eingesetzt. Ein sehr guter, grafisch für den Presentation Manager ausgelegter Talk-Client mit dazugehörigem Talk-Daemon, ist »Gibbon Talk«, das ebenfalls als Shareware vertrieben wird. Dieses Produkt kann unter anderem unter der folgenden ftp-Adresse gefunden und per anonymous-ftp kopiert werden: »ftp://ftp.teamos2.de/pub/OS2/Hobbes/Network/TCPIP/Tools/Talk/gcptlk11.zip«

Dort befindet sich auch ein kommandozeilenorientierter YTalk-Client für OS/2 (ytalkos2.zip).

9.10.10 Drucken im TCP/IP-Netzwerk

Es ist dem Benutzer möglich, über ein auf dem TCP/IP-Protokoll basierendem Netzwerk, Drucker von anderen Computersystemen anzusprechen. Teilweise verfügen einige Netzwerkdrucker auch selber über eine Netzwerkschnittstelle und können so sogar direkt angesprochen werden. Dadurch ist es möglich, Druckaufträge an nicht lokale Drucker zu vergeben. Dabei ist es im Prinzip egal, ob der Drucker neben dem den Druckauftrag versendenden Computersystem steht, oder ob er sich in New York befindet. Über das »Network Printing Protocol« wird dieses ermöglicht. Dabei überträgt ein Druckclient seine Druckdatendatei, im bereits für diesen Drucker aufbereiteten Format, an diesen Druckserver und dieser Server führt den Ausdruck aus.

Unter Warp V4 sind verschiedene Dienstprogramme für diese Aufgaben vorhanden. Diese reichen von den reinen Netzwerkübertragungsclients über Dienstprogramme zur Statusabfrage des entfernten Druckservers bis hin zu einem Druckserver-Daemonprogramm für den lokalen OS/2-Rechner. Dadurch wird es möglich, mit dem OS/2-Rechner einen Druckserver zu betreiben, an den andere Druckclients Druckaufträge verschicken können.

Unter OS/2 Dateien auf einem entfernten Drucker ausgeben. Wie geht das?

Es ist unter OS/2 Warp über das Dienstprogramm »lpr.exe« (»LinePrint Remote«, entfernt Zeilen ausdrucken) möglich, eine Datei auf einem entfernten Computersystem oder auf einen

Netzwerkdrucker ausgeben zu lassen. Auf dem Zielcomputersystem muß ein Drucker-Daemon gestartet sein, der die Druckaufträge entgegen nimmt (dieser Prozeß erwartet Anfragen auf Port TCP 515) und an den entsprechend zum Ausdruck bestimmten Drucker weiterreicht. Handelt es sich bei dem Zielsystem um einen Netzwerkdrucker, so ist diese Funktionalität normalerweise bereits in die Steuersoftware des Druckers integriert.

Durch Aufruf von *lpr -?* wird eine Liste möglicher Kommandozeilenparameter ausgegeben.

Durch den Parameter *-s* oder die Umgebungsvariable *LPR_SERVER* wird der zu benutzende Druckserver definiert. Über diesen Parameter kann auch ein anderer als der über die Umgebungsvariable definierte Druckserver angegeben werden. Auf dem Druckserver muß natürlich ein Druckdaemon gestartet sein, der eingehende Druckaufträge verarbeitet.

Der Drucker, auf dem ein Ausdruck erfolgen soll, wird über die Umgebungsvariable *LPR_PRINTER* oder den Parameter *-p* definiert. Wie auch bei der Auswahl des Servers gilt der Parameter vor der Umgebungsvariable. Der hier definierte Wert muß mit der Angabe der Druckerqueue auf dem Druckserver übereinstimmen, damit eine Ansteuerung erfolgen kann. Im Fehlerfall wird keine Meldung ausgegeben.

Für den Druckerqueuenamen kann ein physikalischer Name wie *LPT1* oder *LPT2* oder ein wirklicher Queuename wie *FxPrint* oder *HPDeskJet* angegeben werden.

Über den Parameter *-b* wird LPR mitgeteilt, daß es sich um eine Übertragung von binären Druckerdatendateien handelt. Dadurch wird eine mögliche Konvertierung von CR/LF ausgeschlossen. Durch *-f* wird eine textbasierende Übertragung angenommen und eventuell eine Ausgabe über *pr* (Unix-Utility) durchgeführt, sofern dieses vorhanden ist.

Der Parameter *-t* schaltet die Rückverfolgung (das Tracing) ein. Dadurch wird der Ablauf der Übertragung stark verlangsamt. Im »temp«-Verzeichnis (wenn die Umgebungsvariable gesetzt wird) wird nun eine Datei mit dem Namen »tracelpr.XXX« angelegt. Dort steht der Inhalt der übertragenen Datei und kann somit als Logfile für spätere Auswertungen dienen. In der Online-Hilfe wird dieser Parameter nicht weiter erwähnt.

Über die Parameter *-q* und *-r* kann definiert werden, wie der Verbindungsaufbau verlaufen soll. Nach wie vielen Sekunden ein erneuter Versuch zur Kontaktaufnahme erfolgen (-q) und wie häufig dieser Versuch unternommen werden soll (-r). Standardmäßig stehen diese Werte auf dreimaliger Verbindungsaufbauversuch und jeweils einer zehn Sekunden dauernden Pause zwischen diesen Wiederholungsversuchen.

Andere vorhandene Parameter werden im normalen Gebrauch fast nie benötigt und daher werden diese hier auch nicht weiter erwähnt.

Kontrolle der Druckjobs durch LPQueue

Durch das Dienstprogramm »lpq.exe« ist es möglich, die Druckerqueue eines entfernten Druckservers zu überwachen und Auskunft über die dort installierten Druckerqueues zu erhalten.

Beim Aufruf dieses DienstProgramms können die Kommandozeilenparameter *-s* den Druckserver und *-p* die Druckerqueue näher spezifizieren, wie es bei den anderen Dienstprogrammen bereits beschrieben wurde.

Befinden sich Druckaufträge in der abgefragten Druckerqueue auf dem Druckserver, so werden nähere Informationen wie Job-ID und Job-Inhaber des Druckauftrages ausgegeben. Diese Angaben können für ein eventuelles Entfernen des Druckjobs mittels des Dienstprogramms LPRM aus der Druckerqueue genutzt werden.

Wird eine Druckerqueue auf einem Serversystem abgefragt, werden nur Angaben zu den Druckaufträgen in dieser abgefragten Druckerqueue gemacht, aber zu Druckaufträgen in anderen Druckerqueues auf dem selben Computersystem werden keine Informationen angegeben.

Wird jedoch der Parameter *-l* angegeben, werden sämtliche vorhanden Druckerqueues auf dem Druckserver aufgelistet, wenn eine Druckerqueue abgefragt wird. Dadurch können unter Umständen Informationen über bisher nicht bekannte Druckerqueues gewonnen werden und diese dann auch abgefragt werden.

Löschen von Druckaufträgen auf entfernten Computersystemen?

Durch das Dienstprogramm »LPRemove« (*lprm.exe*) ist es möglich, Druckaufträge aus der Druckerqueue eines entfernten Druckservers zu löschen, solange nicht bereits mit der Abarbeitung dieses Auftrages angefangen wurde.

Dazu wird die Auftragsnummer (Job-ID) des zu löschenden Druckauftrages benötigt. Diese Angabe kann durch die Verwendung des DienstProgramms »LPQ« ermittelt werden.

Um diesen Druckauftrag nun zu löschen, wird »lprm.exe« mit den entsprechend den Druckserver und die Druckerqueue spezifizierenden Parametern (-s und -p) und der Auftragsnummer als weiteren Parametern aufgerufen. Bei Erfolg wird eine entsprechende Angabe gemacht, daß der Auftrag erfolgreich entfernt werden konnte. Es ist möglich, mehrere Auftragsnummern gleichzeitig durch einen einzigen Aufruf löschen zu lassen. Dazu werden die entsprechenden Daten nur hinten als weitere Parameter angehängt.

Drucken auf entfernte Drucker per Drag&Drop

Natürlich ist es ebenfalls möglich, Druckerdatendateien per Drag&Drop auf entfernte Drucker auszugeben. Dazu wird letztendlich das Programm »lpr.exe« benutzt.

Die Vorgehensweise ist recht simpel. Es werden für jeden entfernten Drucker, der benutzt werden soll, ein Programm-Objekt aus dem Schablonen-Ordner gezogen und auf der Arbeitsoberfläche entsprechend konfiguriert.

Dabei wird als Programmdateiname der Pfad und der Dateiname von »lpr.exe« angegeben (oder einfach nur »lpr.exe«, wenn sich der Pfad, in dem sich das Programm befindet, in der Pfad-Angabe der Umgebungsvariable wiederfindet, was im Normalfall eigentlich der Fall sein sollte).

Als Parameter werden die gewünschten Parameter eingetragen, wobei mit *-p* die gewünschte Druckerqueue auf dem entfernten Computersystem und mit *-s* der Druckservername definiert werden sollte.

Ferner ist die Angabe des Parameters *-b* sinnvoll, da sicherlich nicht nur ASCII-Dateien übertragen werden sollen (Ausnahme eventuell Postscript-Drucker).

Am Ende der Parameterzeile wird nun *%** angegeben. Dadurch wird eine Drag&Drop-Übergabe eines Druckerdatenobjektes ermöglicht.

Was geschieht nun? Das Druckerdatenobjekt wird auf das LPR-Objekt fallen gelassen und wird nun von LPR an den entsprechend definierten Server auf die bezeichnete Druckerqueue gesendet und dort ausgedruckt.

Es ist so möglich, für verschiedene Server mit verschiedenen Druckern mehrere Objekte anzulegen, die die jeweils geforderten Einstellungen bzw. Zieldrucker haben.

Wie erhalte ich jedoch eine Druckerdatendatei? Dazu wird in den Einstellungen des lokalen Druckerobjektes definiert, daß eine Druckausgabe in eine Datei erfolgen soll. Diese Datei (vorrausgesetzt es wurde im Druckerobjekt *Druckerspezifisches Format* gewählt) kann nun auf dem entsprechenden Drucker ausgegeben werden.

Der Druckserver unter Warp V4

Das Programm »lpd.exe« startet den Druckserver. Es können Kommandozeilenparameter übergeben werden, die den näheren Verlauf des Ausdrucks regeln. Über das TCP/IP-Konfigurationsobjekt ist es auch möglich, diesen Dienst automatisch beim Systemstart zu aktivieren. Entweder über den Inet-Daemon-Superserverprozeß oder separat als Vorder- oder Hintergrundprozeß.

Durch die Angabe von *-c* wird ein Ausdruck eines Kontrollberichtes unterbunden, der normalerweise standardmäßig erfolgt. In diesem Kontrollbericht ist zu erkennen, von welchem Benutzer auf welchem Computersystem die Druckanforderung ausgegangen ist, der Name der zu druckenden Datei und eine Identifizierungskennzeichnung.

Wird der Parameter *-b* gefolgt von einem Dateinamen übergeben, erfolgt der Ausdruck dieser Datei als Vorspannseite. Dabei muß es sich bei dieser Datendatei um eine Datei handeln, die in einem für diesen Drucker spezifischen Format oder als ASCII-Datei erstellt wurde. Ist dieses nicht der Fall, wird diese Datei möglicherweise nicht korrekt ausgegeben.

Der Druckserverdaemon unterzieht die eingehenden Dateien *keiner* Umwandlung in das für den angewählten Drucker benötigte Format. Die Inhalte werden direkt weitergereicht. Es muß daher sichergestellt sein, daß der gewünschte Drucker diese Formate auch versteht, d.h., auf dem Ursprungssystem muß der entsprechende Druckertreiber für diesen Druckertypen verwendet werden, damit ein entsprechendes Druckerformat erreicht wird.

In den Druckereinstellungen des die Druckdatei erzeugenden entsprechenden Druckertreibers auf dem Ursprungscomputersystem, muß die Checkbox *Druckerspezifisches Format* unbedingt aktiviert sein, damit der Zieldrucker auch den Inhalt versteht und nicht einfach nur als Binärdateien ausgibt.

Auf der Konsole des LPD ist ersichtlich, von wem eine Druckanforderung gesendet wurde und wie der Dateiname der zu druckenden Datei war, sofern dieser spezifiziert wurde.

Unter OS/2 ist es nicht möglich, nur bestimmten Benutzern die Möglichkeit zu geben, auf diese Druckserverfunktionalität zuzugreifen. Generell ist jeder Benutzer in der Lage, seinen Druckauftrag auf diesem OS/2-Server zu starten, wenn er den Servernamen und die Namen der Druckerports oder Druckerqueues kennt. Daher ist dieses mit ein wenig Vorsicht zu genießen, weil praktisch keine Sicherheit besteht.

Umleitung der Druckausgabe auf entfernte Computer

Vorweg eine kleine Erläuterung zum Unterschied zwischen »LPRPortd« und »LPRMon«. Die Unterschiede sind recht einfach zu erklären. »LPRMon« kann nur die Ausgabe umleiten, die auf die parallelen Ports erfolgen soll. Dieses ist statisch, d.h., eine Umkonfiguration im laufenden Betrieb ist nicht möglich. Es muß erst der LPRMon beendet, dann durch Kommandozeilenparameter neu konfiguriert und danach dann erneut gestartet werden. Außerdem ist so die Umleitung von maximal drei Ports möglich, weil OS/2 nur LPT1, LPT2 und LPT3 standardmäßig kennt (es gibt Ausnahmen, doch die bleiben hier unerwähnt). Mit »LPRPortD« hingegen sind bis zu maximal 64 Umleitungen realisierbar, die Konfiguration ist im laufenden Betrieb veränderbar (über die Einstellungen der Objekte) und die lokalen Druckerports können zusätzlich genutzt werden, was bei LPRMon dann natürlich nicht der Fall ist.

Es ist unter der DOS- und Windows-Emulation jedoch nicht möglich, auf diese neuen durch »LPRPortd« bereitgestellten Druckerports zuzugreifen. Dazu wird jetzt jedoch »LPRMon« genutzt. Wenn unter der Windows-Emulation von OS/2 als Druckerport »LPT1.OS2« angegeben wird und unter OS/2 mit »LPRMon« auf LPT1 eine Umleitung konfiguriert wurde, erfolgt die Umleitung zu dem konfigurierten Druckserver und der Druckauftrag wird ebenfalls auf dem Zielcomputersystem gedruckt, wie es unter der Verwendung von »LPRPortd« mit OS/2-Applikationen der Fall wäre. Dadurch ist auch erklärt, warum beide Dienstprogramme mitgeliefert werden.

Über das TCP/IP-Konfigurationsobjekt kann definiert werden, ob der »LPRPortd« beim Systemstart gestartet werden soll. Der Start über den »inetd« ist natürlich nicht sinnvoll, daher kann diese Option auch nicht ausgewählt werden. Zu Empfehlen ist hier die Option *Hintergrundsitzung*, da so das Programmfenster nicht in der Fensterliste erscheint und die Ausgabe des Programms im Regelfall nicht weiter interessiert.

LPRMonitor

Durch das Dienstprogramm »lprmon.exe« wird es lokalen Benutzern ermöglicht, statt dem lokalen parallelen Druckerport einen entfernten Druckserver anzusprechen. Das Programm fängt eine Ausgabe auf den angegebenen Druckerport ab und sendet die Daten stattdessen auf den Druckserver und an die definierte Druckerqueue.

Dem Programm werden beim Aufruf über Kommandozeilenparameter übergeben, welcher lokale Druckerport überwacht werden, auf welchen Drucker *-p* und auf welchen Druckserver *-s* diese Umleitung erfolgen soll.

Ferner erfolgt wieder die Angabe, ob es sich dabei um Text- oder Binärübertragungen handelt.

Der Verbindungsaufbau kann über die Parameter *-r* (Wiederholungsversuche) und *-q* (Zeitraum zwischen den Wiederholungsversuchen) näher konfiguriert werden.

Es wäre so auch möglich, eine Umleitung von einem lokalen Druckport auf einen anderen lokalen Druckerport zu erstellen. Dieses ist einfacher allerdings auch über die normalen Druckerobjekte lösbar.

LPRPortDaemon

Wird dieser Daemon durch »lprportd.exe« gestartet, werden durch dieses Dienstprogramm standardmäßig acht (jedoch sind generell bis zu 64 möglich) zusätzliche virtuelle Druckerports (durch »Named Pipes«) angelegt. Diese können im Druckerobjekt hinzugefügt werden: *Einstellungen* des Objektes öffnen, *Ausgabeanschluß|neuen Anschluß installieren*, selektieren von *lpd0* bis *lpdX*, *Installieren* anklicken) und dann als Ausgabeanschluß anwählen.

Die Konfiguration eines solchen virtuellen Ports erfolgt durch das Öffnen der Einstellungen des Druckerobjektes und danach durch das Öffnen der Einstellungen des *\pipe\lpdX*-Ausgabeports, wobei dem X die entsprechende Zahl zuzuordnen ist. Dort können unter anderem die Angaben zu Druckserver, Ausgabequeue, Ursprungscomputersystem und Benutzername eingestellt werden.

Dadurch ist es möglich, ohne große Umwege die Druckausgabe direkt auf einen entfernten Drucker umzuleiten, ohne umständlich eine Konfiguration mit »lpr« und »lprmon« zu erstellen. Dieses kann durch jedes beliebiges OS/2-Programm genutzt werden, nicht jedoch durch DOS- oder Windows-Programme in den Emulationen.

Wird »lprportd.exe« gestartet, können auch hier Parameter übergeben werden, die den Netzwerkübertragungverlauf näher spezifizieren. Durch die Angabe des Parameters *-?* wird eine Übersicht über diese Parameter ausgegeben.

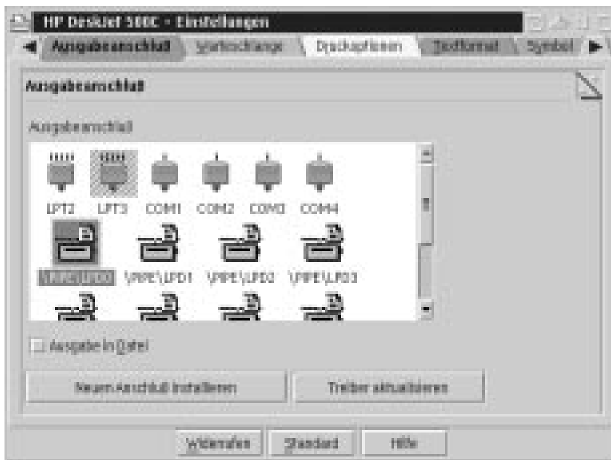


Abb. 9.63: Druckerobjekteinstellungen



Abb. 9.64: Einstellungen der Ausgabeports

9.10.11 Namen und Nummern. Wie funktioniert das eigentlich?

Im Internet werden (aussagekräftige) Namen für Computersysteme benutzt, da die wenigsten Benutzer in der Lage sind, die IP-Nummern von hunderten von verschiedenen Computersystemen im Kopf zu behalten. Aber das IP-Protokoll kann nun nicht nach Namen routen oder auf diesen aufbauen, sondern benötigt dafür die IP-Nummern, die zu dem entsprechenden Computersystem zugeordnet werden.

Diese Aufgabe, das Zuordnen von Namen zu IP-Nummern oder umgekehrt das Zuordnen von IP-Nummern zu Namen, übernimmt ein sogenannter »Domain-Name-Server« (Domänennamensserver), auch als »DNS« bezeichnet. Wie viele es davon im Internet gibt, ist wohl nicht recht bekannt. Um eine Domäne (Domain) im Internet zu betreiben, also inklusive der Informationen über einzelne Computersysteme innerhalb dieser Domain, wird ein Domain-Name-Server benötigt, der diese entsprechenden Daten bereithält und auf Anfrage entsprechend Auskunft gibt. Es wird zwischen *primären* (*primary*) und *sekundären* (*secondary*) Domain-Name-Servern unterschieden. Der sekundäre erhält die Informationen, für die er verantwortlich ist, vom primären Domain-Name-Server. Warum zwei Domain-Name-Server? Falls einer ausfallen sollte, bleibt so ein Backup-Name-Server vorhanden und außerdem ist es normalerweise der Fall, daß sich der primäre und sekundäre Nameserver in verschiedenen Netzbereichen befinden sollten, um so Anfragen von verschiendenen Netzbereichen einigermaßen ausgeglichen schnell beantworten zu können.

Wie funktioniert ein Domain-Name-Server? Ein kleiner Überblick

Ein Domain-Name-Server erhält eine Anfrage über einen Computersystem- oder Hostnamen. Er schaut als erstes in seiner eigenen Datenbank nach. Ist der Domain-Name-Server primary oder secondary für diese Domain, werden diese Anfragen sofort mit den entsprechenden Daten beantwortet.

Ist dies nicht der Fall, wird eine Anfrage an den Root-Nameserver der entsprechenden Top-Level-Domain geleitet. Dieser weiß, ob es diese Domain überhaupt gibt und kann Auskunft über die für diese Domain verantwortlichen Nameserver geben. Der Domain-Name-Server fragt dann bei diesen Nameservern nach den gewünschten Daten. Die dann erhaltenen Daten werden bei diesem Domain-Name-Server dann für einen gewissen Zeitraum (dieser wird im entsprechenden Domain Name-Server-Konfigurationsfile dieser Domain definiert) im Cache gespeichert, so daß nachfolgende Fragen schneller beantwortet werden können.

Sollte zwischenzeitlich der Eintrag auf dem primary Domain-Name-Server verändert werden, werden diese Informationen erst dann auch auf diesem System gültig, wenn entweder der Cache gelöscht wird oder die Lebensdauer des Eintrags im Domain-Name-Server-Cache erreicht wird (welcher Fall auch immer eher eintritt). Solange dies nicht der Fall ist, liefert der Domain-Name-Server die alten und dann sogar ungültigen Daten zurück. Im Normalfall werden die Daten nach ein paar Tagen aus dem Cache entfernt.

Ist ein Domain-Name-Server nicht in der Lage, den Computersystemnamen in dieser Domain aufzulösen (zu »resolven«), wird eine entsprechende Fehlermeldung an das die Daten anfordernde Programm zurückgeliefert. Daß ein Domain-Name-Server nicht in der Lage ist, eine Anfrage nach einem Computersystem in einer Domain zu resolven, kann mehrere Ursachen haben. Von einem Ausfall der Datenleitung in das Internet des Netzes, in dem sich der benutzte Domain-Name-Server befindet, über einen Leitungsausfall des Netzes, in dem sich der für die gesuchte Domain verantwortliche Domain-Name-Server befindet bis hin zu dem Fall, daß es diese Domain nicht gibt. Ist die Verbindung zu den für die Domain verantwortlichen Domain-Name-Servern unterbrochen, existiert diese Domain für den Rest des Internets unter Umständen nicht mehr.

Was alles an Name-Server-Anfragen auf einem unter dem Betriebssystem Unix laufenden Domain-Name-Server eingehen, kann mit Hilfe des Unix-Tools »tcpdump« mittels »tcpdump port domain« kontrolliert werden.

Wie machen TCP/IP-Dienstprogramme diese Konvertierung?

Wenn ein Dienstprogramm wie »Ping« oder »Traceroute« einen Namen als Parameter bekommt, muß erst eine Umwandlung dieses Namens in eine IP-Adresse erfolgen, bevor diese Programme ihren eigentlichen Dienst beginnen können. Sie senden, bildlich betrachtet, eine Anfrage an einen Domain-Name-Server auf Port 53 mit dem Namen als Parameter und erhalten im Gegenzug die dazugehörige IP-Adresse. Diese können die Programme nun verwerten und erfüllen ihren Dienst.

Das Ganze geschieht für den Benutzer des Programms völlig transparent. Das bedeutet, er merkt von dieser Konviertierung nichts, solange diese auch funktioniert. Tritt ein Fehler auf, zum Beispiel weil es diese Domain nicht gibt, so erhält der Benutzer eine entsprechende Fehlermeldung.



Abb. 9.65: Fehlermeldung wegen einer nicht existierenden Domain

Wenn es massive Probleme mit dem Auflösen von Namen zu IP-Adressen gibt, es vielleicht überhaupt nicht mehr funktioniert, so sollte das Vorhandensein der Datei »RESOLV« im »etc«-Verzeichnis (Umgebungsvariable beachten) überprüft werden. Der Name für diese Datei kann unter Umständen auch variieren, bei machen Systemen hat sie den Namen »RESOLV2«. In dieser Datei stehen die notwendigen Daten, um einen Nameserver anzusprechen. Sie enthält den eigenen Domainnamen und eine oder mehrere IP-Nummern von Domain-Name-Servern.

NameServerLookup, das Dienstprogramm »nslookup.exe«

Über dieses Dienstprogramm ist der Benutzer selber in der Lage, verschiedene Informationen zu Einträgen eines Domain-Name-Servers oder einer Domain zu erhalten. Mit diesem Programm kann fast alles über eine Domain in Erfahrung gebracht werden, was öffentlich zugänglich ist. Dieses reicht von der Angabe der für diese Domain verantwortlichen Domain-Name-Server über die Informationen zum Email-Austausch bis hin zu freiwilligen Hardwareangaben einiger Systeme.

Startet man »nslookup« mit der Option *-?*, erhält man die Liste der möglichen Parameterangaben.

Durch Aufruf von »nslookup computer.system.name« wird versucht, über die Benutzung des standardmäßig durch die Datei »RESOLV« definierten Domain-Name-Servers, die IP-Nummer dieses Computersystems herauszufinden. Ist dieses nicht möglich, so kann das oben aufgeführte Gründe haben. Ist die Suche erfolgreich, so wird die IP-Adresse des Computersystems ausgegeben.

Soll ein anderer als der über die in »resolv« definierte Domain-Name-Server zum Abfragen benutzt werden, erfolgt dieses über den Aufruf »nslookup computer.system.name neuer.dns.name«.

Diese Aufrufe geben jeweils nur eine IP-Nummer aus und sind nicht interaktiv. Um in den interaktiven Modus zu schalten, wird »nslookup« ohne jegliche Parameter aufgerufen.

Es erscheint ein Prompt, an dem Befehle eingegeben werden können. Es wird dort auch angegeben, welcher Domain-Name-Server standardmäßig für die Anfragen benutzt wird. Durch die Eingabe von »help« oder »?« erscheint eine Liste der zulässigen Anweisungen.

Die nützlichsten und am häufigsten gebrauchten Befehle oder Einstellungen werden im folgenden grob erläutert.

set querytype definiert die Art der Abfrage.

Häufig benutzte gültige Werte für diese Einstellung sind *a*, *any*, *cname*, *ns*, *ptr*, *mx* und *soa*. Als Abkürzung kann *set q=<Wert>* verwendet werden.

A ist der Standardwert und steht für »Adresse«. Es wird hier lediglich eine IP-Adresse zurückgeliefert, sofern für dieses Computersystem eine solche vorhanden ist und vom Name Server ermittelt werden kann.

ANY liefert in den meisten Fällen nur Informationen über die verantwortlichen Nameserver, analog dem Wert *NS*. Ist der befragte Domain-Name-Server kein primary oder secondary für die abgefragte Domain, so wird die Angabe *Authoritative answers can be found from:* gefolgt von einer Liste von authoritativen Nameservern ausgegeben, die für die entsprechende Domain als primary und secondary Domain-Name-Server fungieren. Wenn entsprechende Daten für das befragte Computersystem vorhanden sind, werden alle hier aufgelistet. Sozusagen als eine Kombination aus mehreren Querytype-Parametern.

CNAME gibt als Ergebnis den eigentlichen Namen für dieses Computersystem aus, sofern der befragte Computersystemname nur ein Alias-Name ist. Außerdem werden auch hier Informationen über die verantwortlichen Domain-Name-Server gemacht.

NS liefert nur die Angaben zu den verantwortlichen Domain-Name-Servern der befragten Systeme zurück. Ist der befragte Domain-Name-Server nicht authoritativ für diese Domain, so werden Angaben zu den authoritativen Domain-Name-Servern ausgegeben.

PTR ermöglicht eine Abfrage des Reverse-Mappings für ein Computersystem. Reverse-Mapping bedeutet, daß die Auflösungsrichtung umgekehrt und versucht wird, zu einer IP-Adresse einen Hostnamen zuzuordnen. Daher werden für Angaben bei *q=PTR* nur IP-Adressen korrekt verarbeitet.

MX liefert Informationen über die für den Mailaustausch mit dieser Domain verantwortlichen Computersysteme. In der Regel sollten für ein Computersystem mehrere *M*ail-e*X*changer vorhanden sein. Dieses sind Computersysteme, die in der Lage sind, Mails für das eigentliche Computersystem zu empfangen, zu verarbeiten und entsprechend der Konfiguration korrekt zuzustellen. Haben Systeme keine MX-Einträge (*MX-Records*), kann es zu Problemen mit der Mailauslieferung an diese Computersysteme kommen. Es kann passieren, daß ein Mail überhaupt nicht zugestellt werden kann. Daß mehrere MXe vorhanden sind, ist auch aus der Sicht sinnvoll, daß, wenn das eigentliche Computersystem nicht erreichbar ist (durch einen System- oder Datenleitungsausfall), die sogenannten Fall-Back-MXe weiterhin die Mail entgegen neh-

men können und dann, sobald das System wieder verfügbar ist, die Zustellung erfolgt. Die Zuordnung, in welcher Reihenfolge welches Computersystem die Mail nun entgegennehmen soll, wird durch Präferenzen geregelt. Es wird zuerst versucht, die Mail an den MX mit der niedrigsten Präferenz zuzustellen. Kann das nicht erfolgen, wird der Reihe nach versucht, die Mail an den nächsthöheren präferierten MX auszuliefern. Sind keine MX-Records vorhanden, wird versucht, die Nachricht an die eventuell vorhandene IP-Adresse des Zielcomputersystems zuzustellen. Kann auch dieses nicht geschehen, wird eine entsprechende Fehlermeldung erzeugt und an den Ursprungsautor geschickt.

»SOA« steht für »Start of Authority«, Begin der Autorität. Dadurch ist ersichtlich, was die für diese Domain verantwortlichen Domain-Name-Server sind (*nameserver*=-Angaben), wer der Ansprechpartner per Email im Domain-Name-Server-Fehlerfall ist (*mail addr* nach der Zeile »Origin«, wobei der erste Punkt durch einen Klammeraffen (@) ersetzt wird), zu welchem Zeitpunkt die Einträge im Domain Name-Server-Datenfile erstellt oder geändert wurden (*serial*) und wie häufig ein Erneuern der Informationen vonstatten gehen soll (*refresh, retry, expire* und *minimum ttl*).

```
> teamos2.de
Server:  ecrc.de
Address:  141.1.1.1

teamos2.de
        origin = ns.POP.DE
        mail addr = noc.POP.DE
        serial = 96061702
        refresh = 28800 (8 hours)
        retry = 7200 (2 hours)
        expire = 604800 (7 days)
        minimum ttl = 86400 (1 day)
teamos2.de      nameserver = ns.hamburg.POP.DE
teamos2.de      nameserver = ns.braunschweig.POP.DE
teamos2.de      nameserver = ns.space.net
ns.hamburg.POP.DE       internet address = 193.98.9.5
ns.braunschweig.POP.DE  internet address = 194.77.61.6
ns.space.net    internet address = 194.59.182.1
```

Dieses ist ein querytype=SOA-Beispiel für die Domain TeamOS2.de. Der Server ist der Domain-Name-Server des ECRC in München.

Der Befehl *ls*

Diesen Befehl braucht man, um sämtliche in einer Domain vorhandenen Computersysteme aufzulisten. Der Aufruf erfolgt durch die Eingabe von »ls domain.name«. Es werden Angaben zu den Domain-Name-Servern gemacht und es werden alle in dieser Domain vorhandenen Computersysteme namentlich mit IP-Nummer aufgefuehrt.

Es ist jedoch nicht bei jeder Domain eine solche Abfrage möglich. Auf dem entsprechenden Domain-Name-Server muß dieses konfiguriert sein, daß solche Abfragen gültig sind.

Unter Verwendung von weiteren Parametern für *ls* lassen sich die gewünschten Informationen noch weiter unterteilen.

9.10.12 Wer, wie oder was bin ich eigentlich?

Unter Warp V4 gibt es mehrere Programme, die in der Lage sind, mehr oder weniger umfassend Auskunft über bestimmte Daten zu einzelnen Computersystemen zu geben. So zum Beispiel auf die Fragen »Wie heißt eigentlich mein Computersystem im Internet?« oder »Welche IP-Nummer habe ich doch gleich?«.

Alles kein Problem, die Dienstprogramme »host.exe«, »hostid.exe« und »hostname.exe« geben darüber und über einiges anderes Auskunft.

Host-ID, welche IP-Nummer habe ich doch gleich?

Durch einfaches Aufrufen von »host.exe« gibt dieses Programm die IP-Nummer des OS/2-Rechners aus, mit der es initialisiert wurde. Übergibt man den Parameter *-?* beim Programmstart, erhält man die Parameterliste. Es besteht die Möglichkeit, eine Host-ID zuzuweisen. Wird dies getan, wirkt sich dies zumindest nicht auf die benutzte IP-Adresse des Computersystems aus.

HostName, wer bin ich?

Durch den Aufruf von »hostname.exe« wird der eigene Computersystemname ausgegeben. Parameter für dieses Programm sind nicht vorhanden.

Welcher Host bist denn Du?

Durch dieses Dienstprogramm ist der Benutzer in der Lage, ähnlich wie bei »nslookup« die IP-Nummer für ein Computersystem in Abhängigkeit eines übergebenen Namens durch einen Domain-Name-Server zu erfahren. Gibt der Benutzer als Parameter jedoch eine IP-Nummer ein, wird über ein Reverse-Lookup die Zuordnung zu dieser IP-Nummer gesucht.

9.10.13 Ping

Ping dient dem Überprüfen der Erreichbarkeit von Computersystemen im Internet. Dies hat einen Grund:

Aufgrund des dezentralen Aufbaus des Internets über vielfältig verschiedene Datenkommunikationsleitungen kann es vorkommen, daß ein Computersystem oder ein ganzer Netzbereich aufgrund eines Ausfalls eines dieser Routingwege (Leitwege) nicht erreichbar ist.

Das Kommandozeilen-Dienstprogramm »ping.exe«

Das Textmodus-Dienstprogramm »ping.exe« im »bin«-Verzeichnis des »mptn«-Verzeichnisses ist im Normalfall in der Lage, herauszufinden, ob das gewünschte Computersystem im Internet erreichbar ist. Das Programm kann jedoch nur Erfolg haben, wenn es kein Computer-

system oder Routersystem gibt, das die von »Ping« verwendeten ICMP-Datenpakete herausfiltert (»Firewall«).

Der Aufruf von »Ping« ist einfach. Dem Programm wird der zu überprüfende Computersystemname oder die IP-Nummer des Systems übergeben. Das Programm sendet dann ein ICMP-Datenpaket an dieses System und erhält im Erreichbarkeitsfall ein anderes ICMP-Datenpaket als Ergebnis zurückgesendet. Aufgrund dieser Information ist »Ping« in der Lage, die Laufzeit zu diesem System zu berechnen. Die Laufzeit wird in Millisekunden angegeben.

```
[G:\]ping www.teamos2.de
PING www.teamos2.de: 56 Datenbyte
64 Byte von 194.77.105.14: icmp_seq=0. Zeit=90. ms
64 Byte von 194.77.105.14: icmp_seq=1. Zeit=60. ms
64 Byte von 194.77.105.14: icmp_seq=2. Zeit=70. ms
64 Byte von 194.77.105.14: icmp_seq=3. Zeit=60. ms
64 Byte von 194.77.105.14: icmp_seq=4. Zeit=190. ms
64 Byte von 194.77.105.14: icmp_seq=5. Zeit=160. ms
```

Erhält »Ping« keine Datenpakete zurück, hat es den Anschein, als ob das Programm nichts tun würde, es gibt keine Ausgabe auf dem Bildschirm. Wird hingegen das Programm beendet, wird eine Paketstatistik ausgegeben, aus der ersichtlich ist, daß keines der ausgesendeten Datenpakete das Ziel erreicht zu haben scheint.

```
[G:\]ping www.teamos2.org
PING www.teamos2.org: 56 Datenbyte

————www.teamos2.org PING Statistik————
100 Pakete übertragen, 0 Pakete empfangen, 100% Paketverlust
```

Obige Pingstatistik wurde über den Aufruf von »ping www.teamos2.org 56 100« erreicht. Die erste Zahl hinter dem Namen des Computersystems gibt die Größe der versendeten ICMP-Datenpakete an, die zweite Zahl definiert die Anzahl der Datenpakete, die gesendet werden sollen.

```
[G:\]ping www.teamos2.de 56 100
PING www.teamos2.de: 56 Datenbyte
64 Byte von 194.77.105.14: icmp_seq=0. Zeit=30. ms
64 Byte von 194.77.105.14: icmp_seq=1. Zeit=120. ms
64 Byte von 194.77.105.14: icmp_seq=2. Zeit=60. ms
...
64 Byte von 194.77.105.14: icmp_seq=98. Zeit=60. ms
64 Byte von 194.77.105.14: icmp_seq=99. Zeit=60. ms

——www.teamos2.de PING-Statistik——
100 Pakete übertragen, 100 Pakete empfangen, 0% Paketverlust
Umlauf (ms)  min/durchschn./max = 60/178/910
```

Neben der Paketstatistik wird auch eine Paketlaufzeitsstatistik ausgegeben, die die minimale, die maximale und die durchschnittliche Paketlaufzeit angibt.

Sollte ein Computersystem keine Datenpakete zurückliefern, kann mit dem Dienstprogramm »tracerte.exe« nach den Ursachen gesucht werden.

PMPing – Ping auf dem PM

Mit Warp V4 wird mit »PMPing« auch eine PM-Applikation »pmping.exe« im »bin«-Verzeichnis des »tcpip«-Verzeichnisses mitgeliefert, die prinzipiell die gleiche Funktion wie die Textmodusversion übernehmen kann. Bei »PMPing« ist es sogar möglich, mehrere Computersysteme gleichzeitig hinsichtlich der Erreichbarkeit zu überprüfen und diese Überprüfung periodisch für alle Systeme wiederholen zu lassen, wobei das Intervall zwischen 1, 10 oder 30 Sekunden sowie 1 oder 5 Minuten frei wählbar ist. Wird ein Host nicht erreicht, erscheint eine entsprechende Statusmeldung im Fenster von »PMPing«.



Abb. 9.66: PMPing in Aktion, alle Rechner erreichbar



Abb. 9.67: PMPing in Aktion, TeamOS/2 International nicht erreichbar

9.10.14 Wo geht es lang?

Wie, sie wissen nicht, wo es langgeht? Traceroute zeigt Ihnen den Weg.

Die IP-Paketvermittlung im Internet ist eigentlich recht dynamisch. Das IP-Protokoll sollte dafür sorgen, daß IP-Datenpakete immer die direkteste und somit eigentlich auch schnellste Route zum Zielsystem nehmen. Nur leider gibt es manchmal auch Ausfälle solcher Routersysteme, so daß die IP-Datenpakete diesen Weg nicht nehmen können. Wie löst man so ein Problem? Wie laufen die Pakete? Wo laufen sie lang? Fragen über Fragen. Ein hilfreiches Dienst-

programm unter Warp V4 kann diese Fragen schnell aufklären. Es ist auf jedem System, auf dem die Pakete TCP/IP und MPTS installiert sind, vorhanden. Die wenigsten Anwender wissen es wirklich zu schätzen, sofern sie überhaupt von der Existenz wissen. Dieses Programm liefert doch recht schnell deutliche Aussagen über eventuell vorhandene Netzprobleme.

Ähnlich wie bei dem Dienstprogramm »Ping« schickt Traceroute (die Datei heißt trotz HPFS »tracerte.exe«) ein ICMP-Datenpaket auf die Reise zum Zielcomputersystem. Bei jedem Computersystem oder jedem Router, den dieses Datenpaket passiert, verursacht es eine Reaktion. Diese Systeme liefern drei ICMP-Antwortpakete als Ergebnis zurück. Anhand der Laufzeiten kann ermittelt werden, wo zum Beispiel ein Engpaß der Datenleitungen vorhanden ist.

```
[G:\]tracerte www.teamos2.org
 0  moria (XXX.XXX.XXX.XXX)  0 ms  0 ms  0 ms
 1  moria (XXX.XXX.XXX.XXX)  0 ms  0 ms  0 ms
 2  Bourbon.Braunschweig.POP.DE (193.98.11.1)  31 ms  31 ms  31 ms
 3  ipgate.hamburg.pop.de (193.101.63.100)  63 ms  94 ms  63 ms
 4  populus.hamburg.pop.de (193.101.63.250)  93 ms  62 ms  63 ms
 5  cisco.hamburg.pop.de (193.98.9.253)  94 ms  63 ms  93 ms
 6  ecrcpop-gw.hamburg.pop.de (193.101.63.229)  63 ms  94 ms  94 ms
 7  S47-POP.Hamburg.ecrc.net (194.221.252.249)  125 ms  157 ms  156 ms
 8  S41-2048k-hamburg.berlin.ecrc.net (194.221.42.145)  125 ms  188 ms  156 ms
 9  S42-2048k-berlin.muenchen.ecrc.net (193.23.5.1)  94 ms  313 ms  500 ms
10  Munich-EBS.Ebone.NET (193.23.5.98)  250 ms  157 ms  187 ms
11  icm-dc-1-S2/7-1984k.icp.net (198.67.131.225)  281 ms  563 ms  562 ms
12  icm-dc-2-F2/0.icp.net (198.67.131.34)  813 ms  500 ms  500 ms
13  icm-pen-1-H1/0-T3.icp.net (198.67.131.18)  844 ms  562 ms  500 ms
14  sl-pen-2-F4/0.sprintlink.net (192.157.69.9)  375 ms  344 ms  312 ms
15  sl-pen-7-F0/0.sprintlink.net (144.228.60.7)  375 ms  343 ms  344 ms
16  sl-accld-1-S0-T1.sprintlink.net (144.228.67.30)  875 ms  375 ms  437 ms
17  t3-1.toronto.onet.on.ca (206.231.247.237)  375 ms  343 ms  375 ms
18  t1-1.toronto.ican.net (206.231.247.254)  344 ms  281 ms  375 ms
19  161.teamos2.org (206.231.255.161)  500 ms  240 ms  350 ms
```

Obiger Output ist das Ergebnis des Aufrufes von »tracerte www.teamos2.org« von meinem verwendeten Computer. Dort ist ersichtlich, welchen Weg ein IP-Datenpaket von hier zu dem WorldWideWeb-Server des internationalen TeamOS/2 nimmt. Hinter dem Namen des entsprechenden Systems befindet sich die dazugehörige IP-Nummer. Dorthinter befinden sich drei Angaben, aus denen dann die Laufzeiten ermittelt werden können.

Treten statt der Laufzeitangaben Sternchen * auf, kann es sich dabei um einen Paketverlust handeln. Dieses bedeutet, daß das Antwortpaket aus nicht bekannten Gründen am Ausgangsort nicht angekommen ist.

Eine Ursache für solche auftretenden Paketverluste kann ein zur Zeit bestehender Datenleitungsengpaß (Lastspitze) sein. Dann gehen die Datenpakete verloren. Zum anderen ist es denkbar, daß ein Router solche gesendeten ICMP-Datenpakete herausfiltert oder gar nicht erst beantwortet. In den meisten Fällen spricht man dann von *firewalling*, das Computer- oder Routersystem fungiert dann als *Firewall*. Wörtlich übersetzt bedeutet dieses »Feuerwand«. Der Begriff hat seinen Ursprung im Schutz gegen Einbrüche auf Computersysteme. Hacker

verbrennen sich daran die Finger, weil diese Firewall, saubere Konfiguration vorrausgesetzt, undurchdringbar sein und somit einen optimalen Schutz bieten sollen.

Aus der obigen Traceroute-Ausgabe ist ersichtlich, daß es 18 Computersysteme oder Router sind. Im Normalfall spricht man von »Hops« (Sprüngen), da zwei Angaben auch von ein- und demselben System stammen können (zum Beispiel bei verschiedenen Interfaces) und nicht notwendigerweise jede Antwort auch für ein selbständiges System stehen braucht. Der WWW-Server befindet sich in Kanada (an der Endung des ca. 17. Systems zu erkennen).

Paketverluste (selten auftretend) werden in der Regel nicht als Fehler- oder Problemfall bezeichnet. Wenn sich jedoch die Paketverluste häufen (daß eine ganze Zeit lang nur Sternchen kommen), so können auch andere Ursachen in genau diesem Moment eingetreten sein. Ein erneuter Traceroute-Aufruf könnte Klarheit schaffen.

```
[G:\]tracerte 104.122.222.2
 0  moria (XXX.XXX.XXX.XXX)  0 ms  0 ms  0 ms
 1  moria (XXX.XXX.XXX.XXX)  0 ms  0 ms  0 ms
 2  Bourbon.Braunschweig.POP.DE (193.98.11.1)  31 ms  31 ms  31 ms
 3  ipgate.hamburg.pop.de (193.101.63.100)  63 ms  63 ms  94 ms
 4  populus.hamburg.pop.de (193.101.63.250)  218 ms  94 ms  62 ms
 5  cisco.hamburg.pop.de (193.98.9.253)  94 ms  63 ms  94 ms
 6  ecrcpop-gw.hamburg.pop.de (193.101.63.229)  125 ms  250 ms  125 ms
 7  S47-POP.Hamburg.ecrc.net (194.221.252.249)  250 ms  156 ms  250 ms
 8  S47-POP.Hamburg.ecrc.net (194.221.252.249)  282 ms !H  173 ms !H  120 ms
!H
```

Ein möglich auftretender Fall ist, daß es vorübergehend keinen Weg zu einem bestimmten Netzbereich oder Computersystem gibt. Die Ausgabe *!H* macht dieses deutlich und bedeutet »No Route to Host« bzw. »Host unreachable« (kein Leitweg zum Computersystem). Wenn statt *!H* der Ausdruck *!N* stehen würde, hätte es eine ähnliche Bedeutung, nur das hiermit »No Route to Network« oder »Network unreachable« gemeint ist.

Es gibt noch einige andere mögliche Fehlerbezeichnungen, die jedoch sehr selten auftreten (mir persönlich bisher noch nicht). Darüber gibt die Online-Hilfe nähere Auskunft.

9.10.15 Auf der Mauer, auf der Lauer.

Woher kommen und gehen eigentlich die IP-Pakete?

Es gibt unter Warp V4 zwei Dienstprogramme, die es ermöglichen, eine grobe Übersicht über den auf dem OS/2-System auftretenden IP-Datenverkehr zu erhalten. Zum einen handelt es sich dabei um »IPTrace« (»iptrace.exe«), das für die eigentliche Aufzeichnung der IP-Datenpakete verantwortlich ist, und »IPFormat« (»ipformat.exe«), das aus dem von »IPTrace« erstellten Logfile eine lesbare Variante eines Logfiles erstellt.

IPTrace

Durch den Aufruf von »iptrace.exe« wird der IP-Paket-Aufzeichner gestartet. Es kann als Parameter definiert werden, welches Netzwerk-Interface protokolliert werden soll. Im LAN ist

dieses normalerweise *lan0*; wenn eine Verbindung über das SLIP- oder PPP-Protokoll über ein Modem besteht, ist es in den meisten Fällen *sl0* oder *ppp0*. Werden die Parameter nicht angegeben, werden alle vorhandenen aktiven Interfaces protokolliert.

»IPTrace« legt im Verzeichnis, von dem der Aufruf des Programms erfolgte, eine Protokolldatei mit Namen »iptrace.dmp« an. In dieser Protokolldatei werden die Datenpakete erfaßt und gespeichert, zusätzlich IP-Headerinformationen, die Auskunft über den verwendeten Protokollport und andere relevante Informationen geben.

Besteht viel IP-Datenverkehr auf dem System, werden diese Logfiles recht schnell sehr groß und das Protokollprogramm übt so eine nicht unerhebliche Systemlast und Plattenbenutzung aus.

Wird diese Protokoll-Datendatei betrachtet, können Datenblöcke daraus erkannt werden, weil diese nicht verschlüsselt gespeichert werden. Jedoch ist diese Datei im Prinzip völlig unübersichtlich, da diese Datei in einem binären Format aufgebaut ist. Genaue Protokollinformationen können so ohne eine geeignete Aufbereitung nicht entnommen werden. Für die Aufbereitung wird das Dienstprogramm »IPFormat« benutzt.

IPFormat

Mit Hilfe dieses Programms kann eine IPTrace-Protokolldatei in ein halbwegs leserliches Format konvertiert werden. Durch den Aufruf von »ipformat« wird die Protokolldatei »iptrace.dmp«, die sich im aktuellen Verzeichnis befinden muß, geöffnet und konvertiert auf dem Bildschirm ausgegeben.

Um die Ausgabe in eine Datei umzulenken, erfolgt der Aufruf mit den Parametern > *dateiname*. In dieser Datendatei steht jetzt ausführlich, welche Datenpakete über das protokollierte Interface gegangen sind.

In dieser Datei sind unter anderem Angaben über Ursprungs- und Zielcomputersystem des IP-Datenpaketes enthalten.

Am Anfang wird die IP-Paketnummer angezeigt, unter der dieses Paket registriert wurde. Wird »IPTrace« gestartet, erhält das erste mitprotokollierte IP-Datenpaket die laufende Nummer 1, das zweite IP-Datenpaket erhält die Nummer 2 und so weiter. Es folgen dann die Angaben zu der IP-Datenpaketlänge, den Ziel- und Ursprungscomputersystemadressen (jeweils die Hardware-Ethernetadresse und die darauf abgebildete IP-Adresse). Anschließend wird der IP-Header weiter aufgeschlüsselt.

Diverse nähere und tiefergehende Angaben über IP-Pakete werden hier an dieser Stelle nicht weiter erläutert, da diese zu weit führen würden. Es gibt dafür ausgezeichnete Bücher, die sich nur mit diesem Thema befassen.

Einige nützliche Informationen wie *Time To Live* (der Zeitraum, wie lange das Paket unterwegs zum Zielsystemsein darf, bevor es von einem Router gelöscht wird), Protocol (hier TCP, andere Möglichkeiten UDP, ICMP, IGMP oder ARP) und *Header Checksum* werden ebenfalls im IP-Header angegeben.

Im TCP-Header dieses betrachteten Datenpakets ist ersichtlich, wo der Ursprungsport der Übertragung und was der Zielport auf dem Zielcomputersystem ist. Der Wert 110 bei der Portangabe gibt an, daß es sich hierbei um eine Übertragung von Email von einem POP3-Mailserver zu einem POP3-Mailclient wie »PMMail« oder »UltiMedia Mail/2 Light« handelt.

Mit dem Dienstprogramm »IPFormat« ist der Benutzer ebenfalls in der Lage, aus der Protokolldatei nur gewisse IP-Datenpakete anzeigen zu lassen. Er hat die Möglichkeit, ARP-, TCP-, UDP-, ICMP- und IGMP-Pakete herausfiltern lassen.

Dieses ist zum Beispiel für die Fehlersuche recht nützlich. Als Beispiel stelle man sich vor, daß ein Benutzer eines OS/2-Computersystems eine ISDN-Wählverbindung in das Internet zu seinem Provider unterhält. Diese Verbindung wird aktiviert (damit ist der Aufbau der Telefonleitung über einen ISDN-Anruf gemeint), sobald ein IP-Datenpaket in das Internet gesendet werden soll. Hat der Benutzer nun unter Umständen den Route-Daemon (»routed.exe«) aktiviert und dieser announciert über »RIP«- (Routing Information Protocol) Routen, kann es vorkommen, daß die Telefonverbindung nicht wieder abgebaut wird, weil zirca alle 30 Sekunden ein RIP-Update vom Route-Daemon gesendet wird. Der Anwender bekommt dieses Problem nur anhand der astronomisch hohen Telefonrechnung mit (natürlich auch dadurch, daß die ISDN-Telefonleitung quasi permanent aufgebaut ist). Die Suche nach der Ursache dieses Problems geht also los.

Der Anwender startet dazu das Programm »IPTrace« und läßt dieses eine Weile den IP-Verkehr mitprotokollieren. Während dieser Zeit unternimmt er keinerlei Aktionen, um keine Datenpakete durch andere Dienstprogramme zu erzeugen, die die Ursachenbestimmung erschweren könnten. »IPTrace« zeichnet die RIP-Pakete (auf UDP basierend) auf, und der Anwender ist hinterher in der Lage, zu sehen, welche IP-Datenpakete für den Leitungsaufbau über ISDN verantwortlich waren. So kann er den Route-Daemon als Ursache identifizieren und wieder deaktivieren und dadurch das Problem lösen.

Um die Logfileauswertung effizienter zu gestalten, können dem Programm »ipformat.exe« diverse Parameter übergeben werden. Die Parameterliste wird durch den Aufruf »ipformat -?« ausgegeben.

Durch den Parameter *-d* werden die IP-Paketinhalte außer dem IP-Header nicht in das konvertierte Logfile übernommen. Dies verkleinert den Platzbedarf des Logfiles.

Über die Angabe des Parameters *-f* kann ein alternatives IPTrace-Logfile zur Auswertung als Eingangsquelle benutzt werden. Standardmäßig wird »iptrace.dmp« aus dem aktuellen Verzeichnis als Logfile zum Auswerten benutzt.

Die Paketfilterung für das Logfile wird über den Parameter *-e* erreicht. Um Datenpakete des Typs *ARP* herauszufiltern, erfolgt der Aufruf mit *-ea*. Analog wird bei der Parameterangabe für die Pakettypen *UDP u*, *TCP t*, *ICMP i* und *IGMP g* verfahren.

Wenn der OS/2-Rechner als IP-Router verwendet oder betrieben wird, gehen natürlich auch die IP-Datenpakete über diese Interfaces, die nicht vom lokalen OS/2-Computersystem, sondern von einem anderen Computersystem erzeugt wurden. Es ist möglich, über den Parameter *-s* gefolgt von der zwölfziffrigen hexadezimalen Ethernethardwareadresse des gewünschten Computersystems zu erzwingen, daß nur Datenpakete mit dieser Hardwareadresse durch

»IPFormat« ausgewertet werden. Dadurch ist es möglich, eine nach Computersystemen geordnete Auswertung der IP-Datenpakete zu erstellen. Anwendungsbeispiele können IP-Accounting oder Durchsatzratenauswertungen sein.

Um herauszubekommen, welcher Dienst welches Protokoll nutzt, kann die Datei »services« im »etc«-Verzeichnis angesehen werden. Dort steht, auf welchem Port der Service Verbindungswünsche erwartet und auf welchem IP-Protokoll (TCP oder UDP) die Übertragung geschieht. So kann also auch konkret nach gewissen Diensten gesucht werden. Leider ist »IPFormat« nicht in der Lage, nach einem bestimmten Port zu filtern.

9.10.16 Der Portmapper

Mit Warp V4 wird ein Portmapper-Dienstprogramm »portmap.exe« mitgeliefert. Dieses Programm ist in der Lage, einen IP-Dienst auf dem lokalen OS/2-Computersystem zu etablieren, der die Auskunft auf Anfragen von Clients über vorhandene TCP/UDP-Ports ermöglicht. Der Daemon wartet auf Port TCP/UDP 111 auf Anfragen.

Durch diesen Dienst ist es möglich, alle zur Zeit von Serverprozessen unterstützten Ports herauszufinden und so Rückschlüsse auf die auf diesem Computersystem betriebenen Daemonen zu erhalten.

Das Programm kann nicht über den Inet-Daemon gestartet werden. Über das TCP/IP-Konfigurationsobjekt kann zwischen den Punkten *Im Hintergrund* und *Vordergrundsitzung* gewählt werden, wobei hier die Option *Im Hintergrund* sicherlich sinnvoller erscheint, sofern nicht durch den Parameter *-d* der Debug-Modus eingeschaltet wurde. In diesem Fall wäre eine eventuell vorhandene Ausgabe auf dem Fenster nicht zu sehen.

9.11 Remote Access Services

von Oliver Mark

Für alle Leser, die schon die vorangegangenen OS/2-Versionen kennen, sei vorab gesagt, daß die Remote Access Services weitestgehend beim LAN-Distance-Client der Warp-Connect-Version entsprechen. Die größte Änderung hat sich im Namen ergeben, ansonsten wurde das Produkt weitgehend unverändert übernommen.

Die Remote Access Services ersetzen einen physikalischen LAN-Adapter durch einen logischen LAN-Adapter, der durch eine Kommunikationsschnittstelle und Software gebildet wird. Es wird mittels serieller Schnittstelle, ISDN-Karte oder einer Co-Prozessorkarte und einer zusätzlichen Protokollschicht ein LAN-Adapter abgebildet, der sich den Netzwerkanwendungen als normaler Adapter präsentiert. Dies ermöglicht den Zugriff über WAN-Strecken auf ein LAN. Der Benutzer wird wie ein lokal angeschlossener Benutzer behandelt. Die Remote

Access Services ersetzen weder den eigentlichen Client (LAN-Requester, Novell-Requester), noch die Terminalsoftware, noch ein Datenübertragungsprogramm, sondern bilden lediglich eine Protokollschnittstelle.

Die Remote Access Services können in verschiedenen Umgebungen eingesetzt werden:

– Remote zu LAN: Die sicherlich weit verbreiteste Variante ist die Anbindung einer entfernten Arbeitsstation an ein LAN. Dies kann ein Heimarbeitsplatz sein, oder ein mobiles Terminal, das sich über Wählleitung in das LAN der Firma einwählt. Dort können LAN-Server-Ressourcen, LAN-Drucker oder auch Großrechnerverbindungen genutzt werden.

– Remote zu Remote: Hierbei werden zwei Remote-Access-Services-PCs untereinander verbunden. Dies kann über Telefonleitung als Datenaustauschsstrecke genutzt werden. Vorstellbar ist auch die Realisierung einer einfachen Peer-Verbindung über eine Haus-Telefonanlage.

– LAN zu LAN: Mittels zwei Remote-Access-Services-Servern kann auch die Verbindung zwischen zwei LANs hergestellt werden. Dies kann zum Beispiel als kurzzeitige LAN-Verbindung dienen, ohne teure Router-Hardware anschaffen zu müssen. Er großvolumiger Datenaustausch setzt aber auch hier die entsprechenden Leitungskapzitäten voraus.

– Remote zu zentralem Server: Im Unterschied zu den vorherigen Alternativen wird hier eine entfernte Arbeitsstation mit einem Server verbunden, der nicht in einem LAN angebunden ist. Im Gegensatz zu *Remote zu Remote* können bei *Remote zu zentralem Server* mehr als ein Benutzer gleichzeitig mit der zentralen Stelle kommunizieren; abhängig von der Anzahl der installierten Modems. Momentan sind Server-Versionen mit bis zu 128-Kommunikationsverbindung verfügbar.

9.11.1 Installation

Die Installation kann, wie bei allen anderen Komponenten, während der Erstinstallation durchgeführt werden, indem der Menüpunkt *Remote Access Services* ausgewählt wird. Oder mittels Netzwerkinstallation anpassen nachinstalliert werden.

Dort können Sie folgende Parameter mitgeben

– *Installationslaufwerk*: Wählen Sie dort ein Laufwerk mit genügend freiem Speicherplatz; empfehlenswert ist, das gleiche Laufwerk wie alle anderen Netzwerkkomponenten zu benutzen.

– *Telefonnummer des Connection-Server*: Wollen Sie Remote Access Services benutzen, um sich an einen RAS-Server anzuwählen, geben Sie dort die Nummer mit Vorwahl ein.

– *Modem-Typ*: Haben Sie ein Modem, das Sie in der Liste wiederfinden, können Sie es dort auswählen, ansonsten belassen Sie es bei dem Standardeintrag *Nicht aufgeführtes Modem*, dies gilt insbesondere auch für die Benutzer von ISDN.

– *COM-Port*: Remote Access Services bietet hier nur die freien seriellen Schnittstellen an, d.h., wenn Sie die physikalischen Schnittstellen des PCs bereits belegt haben, so erscheinen

diese dort nicht mehr. Lassen Sie auch hier den Standardeintrag, wenn Sie ein Schnittstelle auf einer Co-Prozessorkarte oder ISDN benutzen wollen.

– *Type des LAN*: Hier geben Sie an, welches LAN sich hinter dem Connection-Server verbirgt. Hier gibt es nur *Ethernet* oder *andere*, wobei *andere* für Token Ring steht.

Die Installation läuft dann automatisch ab. Nach der Installation finden Sie den Remote Access Client etwas gewöhnungsbedürftig im Connectionsordner unter dem Netzwerkeintrag.

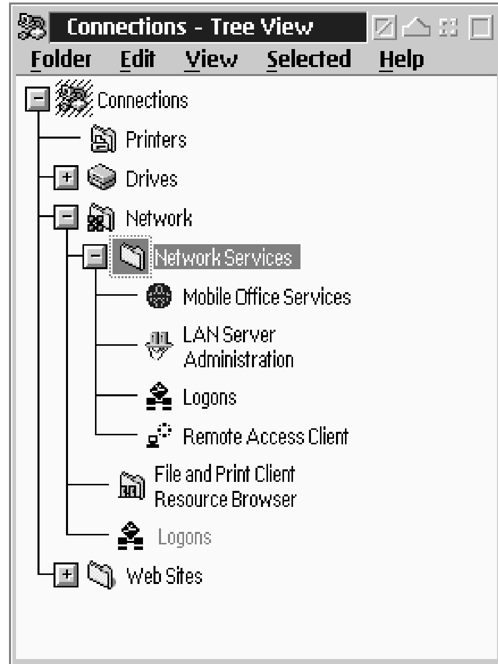


Abb. 9.69: Remote Access Client

Wird der Remote-Access-Services-Client während der Erstinstallation mitinstallatiert, ist dieser im passiven Status. Das bedeutet, daß beim ersten Aufruf des Remote-Access-Clients die Shuttle-Funktion aktiviert, die den Benutzer fragt, ob beim nächsten Booten der PC als Remote-Access-Services-Client oder als LAN-Client agieren soll. Mittels dieser Shuttle-Funktion werden mehrere Dateien ausgetauscht:

– die Config.Sys gegen eine Version mit Remote-Access-Services-Treibern,

– die Protocol.Ini, angepaßt an die Remote-Access-Services-Treiber.

– die Ibmlan.Ini mit angepaßten Timeout-Parametern, um die Netzerkunterstützung auf einer WAN-Verbindung zu ermöglichen.

Probleme mit dieser Funktion bestehen dann, wenn Sie nachträglich in der Remote-Access-Services-Umgebung oder in der LAN-Umgebung eine dieser Dateien ändern; diese Aktuali-

sierungen werden leider nicht in den jeweils inaktiven Backup-Satz übernommen. Sie müssen also eintweder manuell eingefügt werden (aber nur, wenn Sie sich der Sache sehr sicher sind) oder die Remote Access Services müssen deinstalliert werden, die Änderungen vorgenommen und die Remote Access Services wieder installiert werden.

Nach dem Neustart des PCs kann nun die Remote Access Services aktiviert werden.

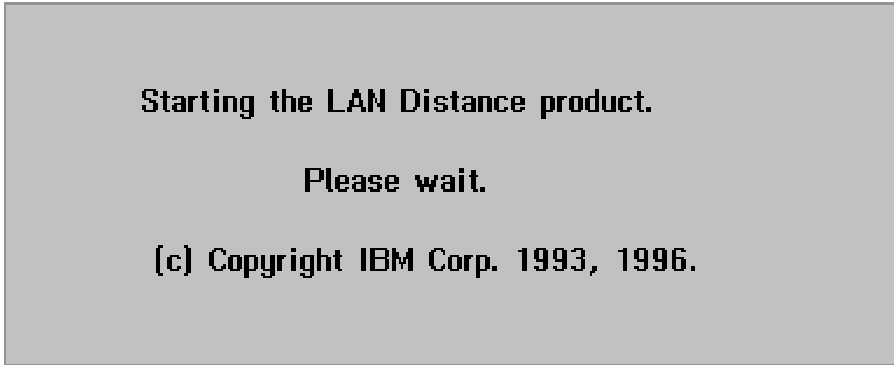


Abb. 9.70: Remote Access Services aktiviert

Hier sieht man deutlich die Verwandtschaft mit dem IBM-Produkt »LAN Distance«. Allerdings wird nun LAN Distance mit Standardeinstellungen gestartet; unter Umständen bekommen Sie auch eine Fehlermeldung, daß das Modem nicht angesprochen werden kann, insbesondere, wenn Sie Nicht-Standard-Modems, ISDN oder Co-Prozessorkarten verwenden wollen. Sollten Sie eine ISDN-Karte oder eine Co-Prozessorkarte verwenden wollen, verzweigen Sie bitte an dieser Stelle in deren Dokumentation zur Installation der LAN-Distance-Unterstützung. Bei der weitverbreiteten Teles S0.16 müssen Sie zunächst die CAPI-Unterstützung installieren, danach die ANDIS-Treiber, die Sie beide aus der Teles-Mailbox erhalten können.

Nun müssen ihre Workstation korrekt konfigurieren. Dazu klicken Sie mit der rechten Maustaste auf das Symbol *Meine Wokstation*.



Abb. 9.71: Das Standardfenster

Das Konfigurationsnotebook wird nun aufgerufen.

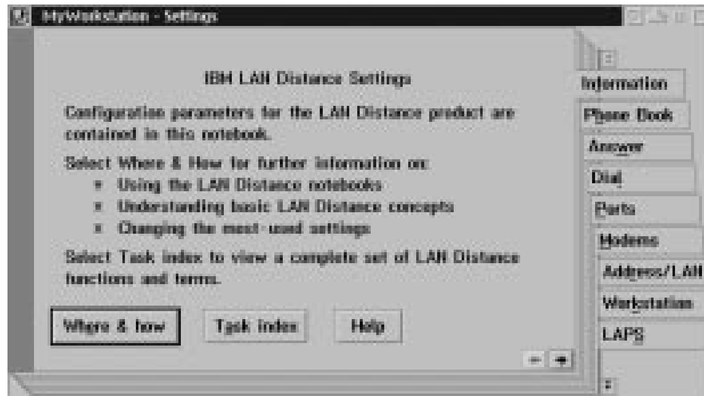


Abb. 9.72: Konfigurations-notebook – Eingangsbild

Von diesem ersten Panel aus können Sie in die Online-Hilfe von Remote Access Services verzweigen. Für die Konfiguration können Sie die einzelnen Zungen des Notebooks direkt anwählen, oder Sie benutzen den kleinen Pfeil rechts unten inneralb der Notebook-Seite.

Die erste Seite beschreibt die Einträge, welche Nummern bei der Benutzung gewählt werden sollen, also die Verbindung zum Server oder zum zweiten RAS-Client. Wählen Sie hier den Button *Change* aus. Hier können Sie beliebig viele Nummern eintragen, die Sie bei der Anwahl zur Auswahl bekommen und diesen Nummern eine Reihenfolge mitgeben, sodaß Sie zum Beispiel bei einer besetzten Leitung zu einem Server auf eine andere Nummer oder einen anderen Server ausweichen können. Jeder Nummer können Sie auch ein bestimmtes Modem und/oder einen Anschlußport zuordnen. Mittels *Autostart* können Sie nach erfolgreichem Aufbau einer Verbindung ein Programm automatisch starten, zum Beispiel den Logon im LAN oder die Terminalemulation für eine Großrechnerverbindung.

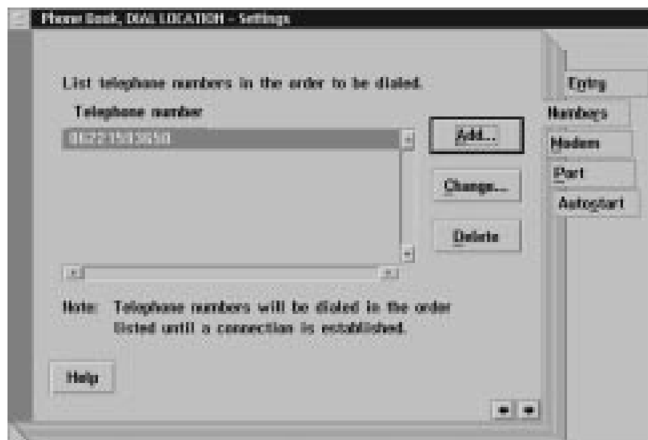


Abb. 9.73: Konfigurationsnotebook – Phone Book

Im nächsten Panel können Sie angeben, ob und in welchem Modus Sie auf eingehende Anrufe antworten wollen. Dies macht nicht nur bei einem Server Sinn, sondern auch in Hinsicht auf die Verbindungsart *Remote zu Remote*. So können Sie automatisch Anrufe von anderen Remote-Clients entgegennehmen. Standardmäßig sind hier alle Dienste gestoppt.

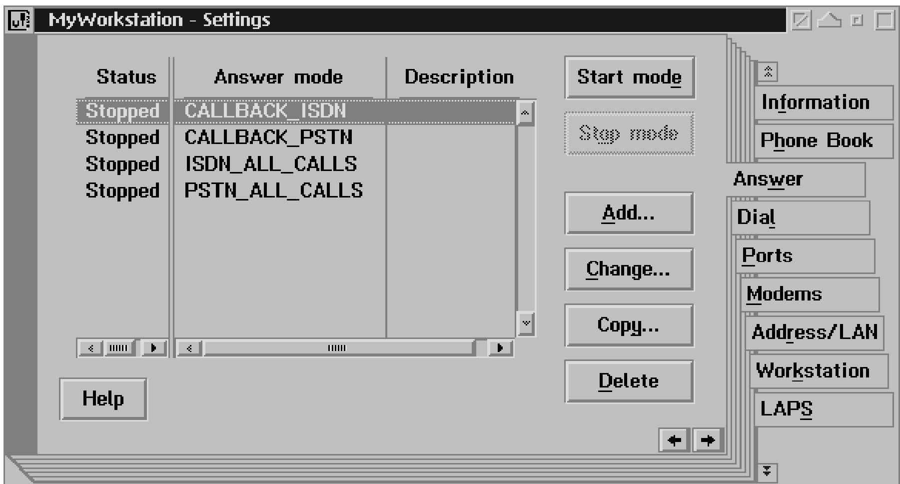


Abb. 9.74: Konfigurationsnotebook – Answer Modus

Ob Sie eine Meldung über einen eingehenden Anruf oder eine unerwartete Unterbrechung eines eingehenden oder ausgehenden Anrufes erhalten wollen, geben Sie im Panel *Dial* ein. Dort können Sie auch definieren, wie oft und in welchen Abständen eine automatische Wahlwiederholung bei besetzter oder nicht erreichbarer Gegenstelle durchgeführt werden soll.

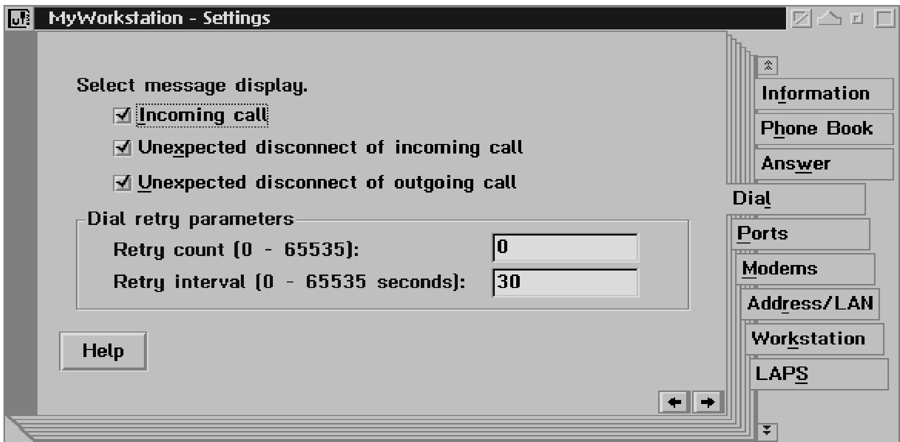


Abb. 9.75: Konfigurationsnotebook – Dial

Auf welchem Kommunikationsport die Remote Access Services-Verbindung durchgeführt werden darf und kann, definieren Sie im folgenden Panel. Haben Sie im Vorfeld ISDN- bzw. Co-Prozessorkarten-Unterstützung installiert, sollten Sie diese Einträge hier angezeigt bekommen. Ansonsten sehen Sie hier ihre lokale seriellen Schnittstellen. Eine Remote-Access-Services-Verbindung über eine serielle LAN-Ressource ist nicht möglich, da Sie damit versuchen würden, eine LAN-Verbindung über eine LAN-Verbindung herzustellen.

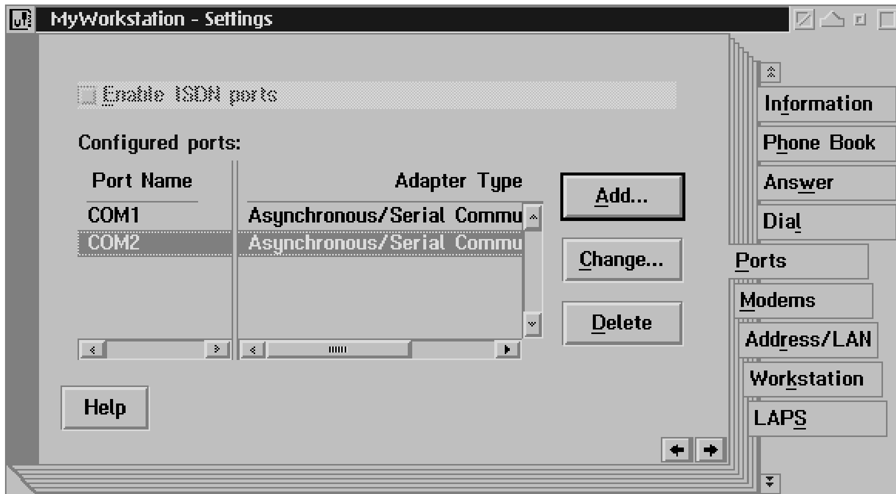


Abb. 9.76: Konfigurationsnotebook – Ports

Die Detailkonfiguration ihres angeschlossenen Modems an den seriellen Schnittstellen können Sie im nächsten Panel vornehmen. Sollten Sie einen häufigen Einsatz der Remote Access Services beabsichtigen, ist der Einsatz von ISDN oder hochwertigen und unterstützten Modems absolut zu empfehlen.

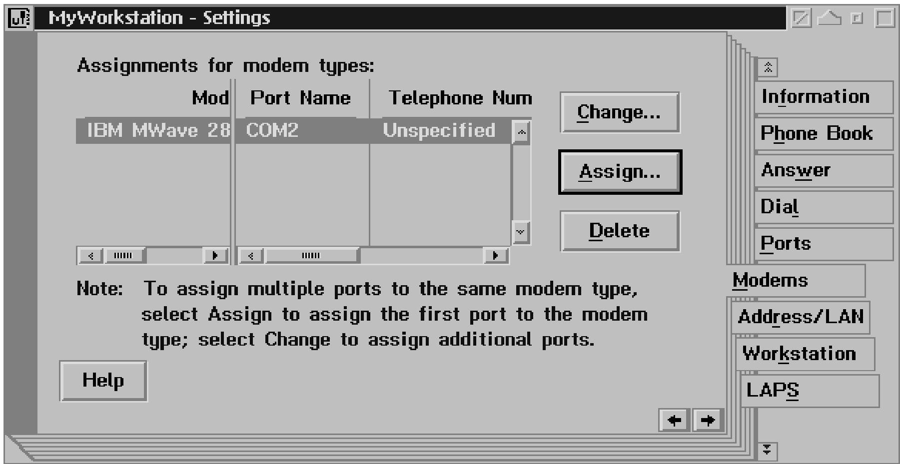


Abb. 9.77: Konfigurationsnotebook – Modems

Mit welcher logischen LAN-Adapter-Adresse sich ihre entfernte Arbeitsstation im LAN anmeldet, können Sie gleichfalls definieren. Bei der Installation wird per Zufallsgenerator eine logische Adapter-Adresse vergeben. Sollten in Ihrem Netzwerk Adapter-Adressen zentral vergeben werden, so tragen Sie diese hier ein. In der Parallele dazu, können auf dem Remote-Access-Services-Server für jeden Benutzer der Zugriff auf eine bestimmte LAN-Adresse beschränkt werden, um so einen Mißbrauch einer Benutzer-ID zu verhindern. Desweiteren können Sie hier definieren, ob das Netzwerk auf der Gegenseite ein Ethernet-Netzwerk oder ein TokenRing-Netzwerk ist.

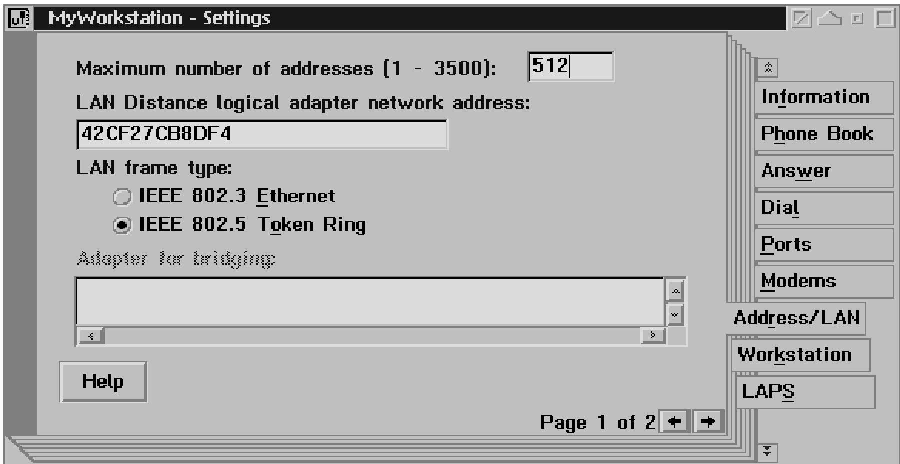


Abb. 9.78: Konfigurationsnotebook – LAN Anschluß

Auf dem Panel *Workstation* können Sie Ihrem PC einen sinnvolleren Namen als *Meine Workstation* vergeben und zusätzlich einen Kommentar einfügen, den Sie sehen, wenn Sie alle RAS-Benutzer eines Servers abfragen.

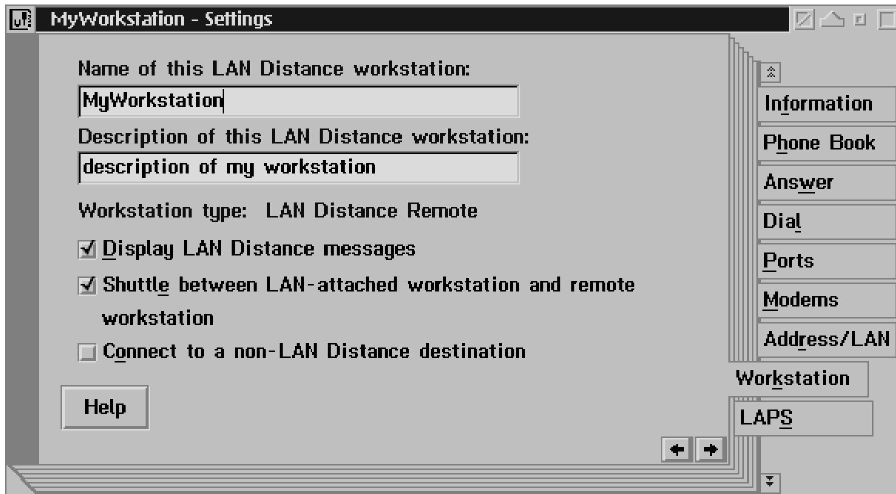


Abb. 9.79: Konfigurationsnotebook – Workstation

Auf der Seite *LAPS* können Sie in den *LAN Adapter Protokoll Support*, oder auch MPTS (Multiprotokoll Transport Services) genannt, verzweigen. Dort können Sie den logischen LAN-Distance-Adapter wie einen physikalischen Netzwerkadapter konfigurieren, Protokolle hinzufügen oder entfernen. Bedenken Sie aber, daß diese Einstellungen dann nur für die Remote-Access-Services-Sitzung gilt, und nicht für den Betrieb als LAN-Client.

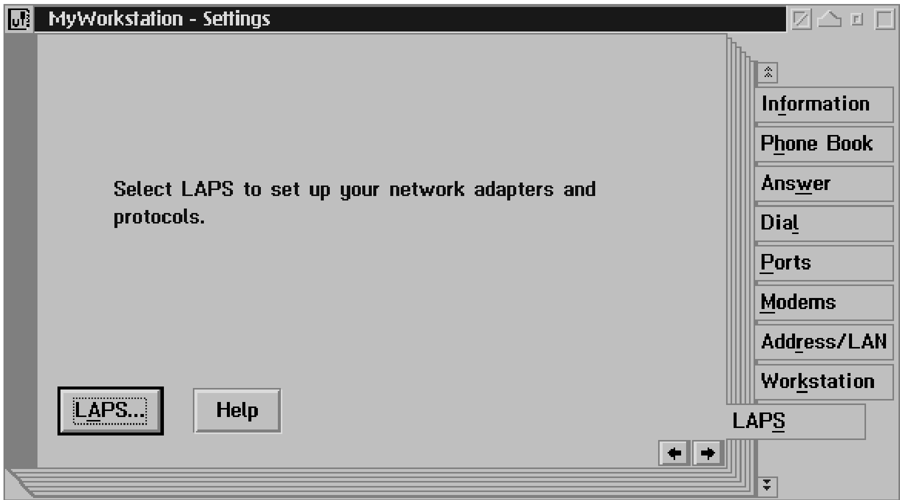


Abb. 9.80: Konfigurationsnotebook – LAPS

Auf der Seite *Timeouts* können Sie definieren, wie groß die Timeoutwerte für die unterstützten Protokolle sein dürfen, bevor ein Fehler auftritt oder die Netzwerkverbindung unterbrochen wird. Hier sollten Sie nichts ändern, solange Sie keine gravierenden Fehler im Betrieb haben. Falls Sie dort Änderungne vornehmen müssen, sollten Sie sich intensiv mit dem IBM-Redbook zu LAN-Distance beschäftigen.

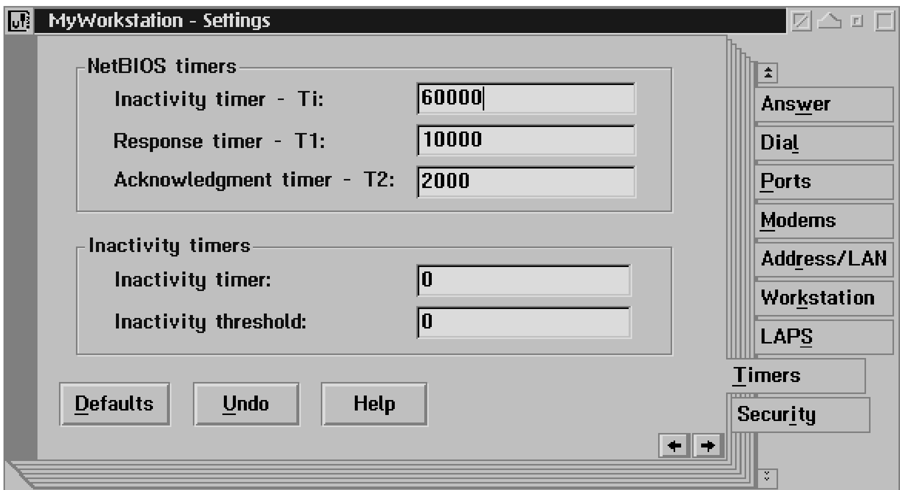


Abb. 9.81:Konfigurationsnotebook – Timeouts

Im letzten Konfigurationspanel schließlich können Sie die lokale Sicherheit für Remote Access Services aktivieren. Das bedeutet, daß Sie sich zunächst lokal an Ihren Remote Access Services-Client anmelden, bevor Sie die Verbindung herstellen können. Im Normalfall sollten Sie das lassen.

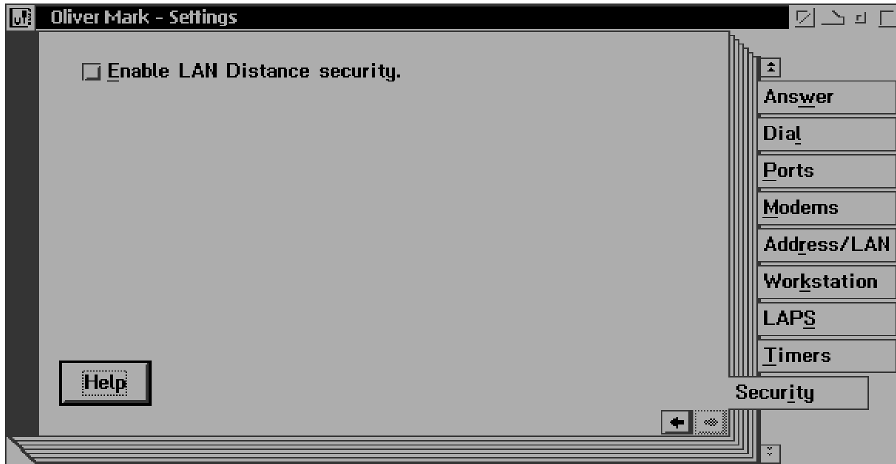


Abb. 9.82: Konfigurationsnotebook – Security

Nachdem Sie nun alle Angaben korrekt durchgeführt haben, müssen Sie nochmals einen Neustart durchführen. Beenden Sie hierfür die Remote Access Services und beantworten Sie die Frage nach der Betriebsart von der Shuttle-Funktion wieder mit *Remote Access Services*.

9.11.2 Deinstallation

Zum Löschen der Remote Access Services sollten Sie zunächst mit der Shuttle-Funktion in der LAN-Betriebsmodus gewechselt haben, um gesperrte Dateien und aktive Konfigurationen zu vermeiden. Das Löschen der Remote Access Services erfolgt druch den Aufruf von *Ldremove* in einem OS/2-Kommando-Fenster. Damit werden alle LAN-Distance-Dateien gelöscht und die Einträge in den Konfigurationsdateien zurückgenommen.

9.11.3 Benutzung

Nach erfolgreicher Installation und Konfiguration können Sie mittels Doppelklick auf Ihr Workstation-Symbol das Anwahl-Fenster starten. Je nachdem, wieviele Kommunikationsmöglichkeiten über verschiedene Modems Sie definiert haben, soviele Seiten erhalten Sie in diesem Fenster.

In der Mitte der jeweiligen Seite sehen Sie die definierten Telefonnummern zur Anwahl. Mittels der Buttons *Wählen* und *Auflegen* steuern Sie die Verbindung. Mit dem Button *Alternate* können Sie zwischen den definierten Nummern für das jeweilige Modem wechseln. Im Fenster *Modem* können Sie den An- und Abwahlvorgang mittels AT-Kommandos verfolgen.

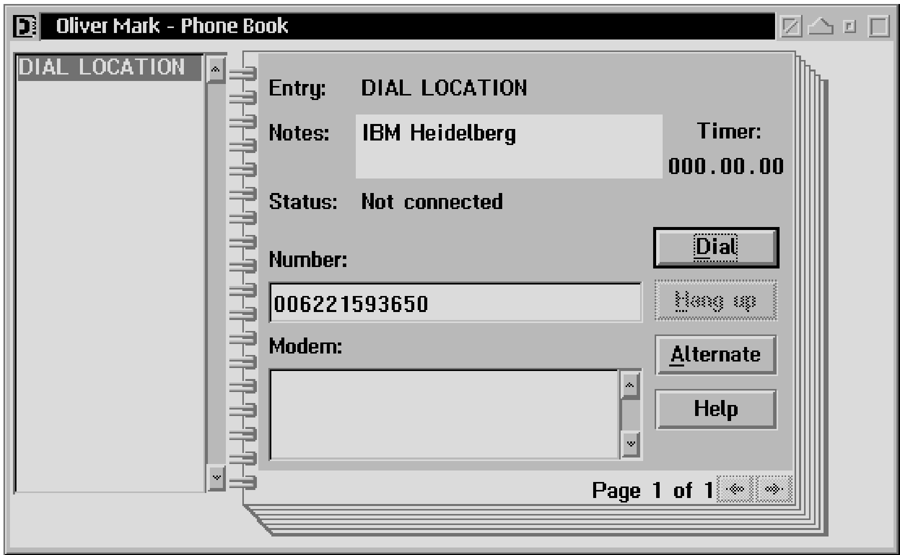


Abb. 9.83: Konfigurationsnotebook – Anwahl

Nachdem die Modem-Verbindung hergestellt wurde, meldet sich der Remote Access Services-Server mit einem Logon-Fenster.

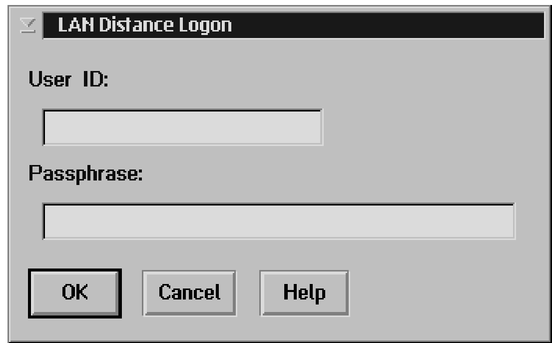


Abb. 9.84: Konfigurationsnotebook – Logon

Beim erstmaligen Logon an einen Remote-Access-Services-Server ist es meist notwendig, eine Paßwortänderung durchzuführen. Dies wird vom Administrator eingestellt. Der Remote-Access-Services-Client meldet sich dann mit der Option, um das Paßwort zu ändern.

Nachdem das Paßwort erfolgreich aktualisiert wurde, ist die Verbindung erfolgreich hergestellt. Für die Verbindung zum Server existieren drei Betriebsmodi: Verbindung ohne Rückruf, Verbindung mit variablem Rückruf und Verbindung mit statischem Rückruf

Bei einer Verbindung ohne Rückruf ist die Sitzung nun aufgebaut und der anrufene Benutzer trägt die Verbindungskosten. Wurde nun auf dem Remote-Access-Services-Server eine variable Rückruffunktion für den Benutzer eingestellt, erscheint ein weiteres Fenster, in dem die Rückrufnummer angegeben werden kann. Dies ist vorteilhaft für mobile Arbeitsplätze, die zum Beispiel vom Kunden oder Hotel aus in der Firma arbeiten wollen. Bei einer statischen Rückrufnummer kann diese Nummer nicht verändert werden. Dies hat bei zum Beispiel festen Heimarbeitsplätzen den Vorteil, daß selbst bei erfolgreichem Logon ein unberechtigter Benutzer von einem anderen Anschluß aus nicht zurückgerufen wird.

Nach dem Logon, respektive dem Rückruf, erscheinen die erreichbaren Remote Access Services-Server.

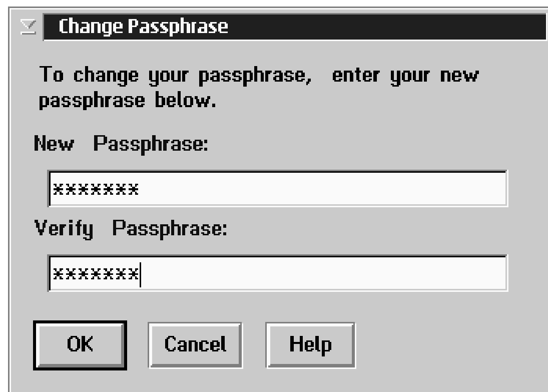


Abb. 9.85: Verbindung aufgebaut, Serveranzeige

Auf den einzelnen Servern kann nun mittels Account-Informationen der eigene Benutzereintrag eingesehen werden und gegebenenfalls das Paßwort geändert werden.

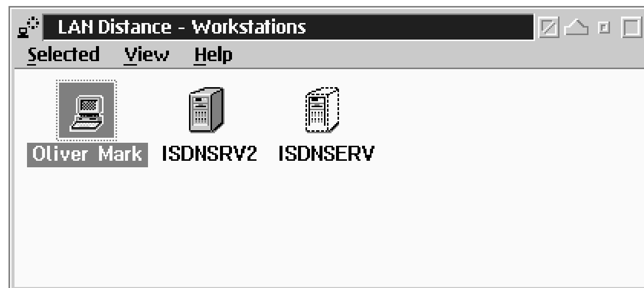


Abb. 9.86: Benutzerinformationen auf dem Server

Auf dem aktuellen Server können zusätzlich verschiedene Informationen abgefragt werden, wie zum Beispiel die zur Zeit angemeldeten Benutzer oder auch die Verbindungsleitungen, die dieser Server zur Verfügung stellt, und deren Stati.

Somit ist die Verbindung komplett transparent für den Benutzer, egal an welchem Ort der Welt, eine brauchbare Leitungsverbindung vorrausgesetzt. Von hier aus kann der Benutzer im

LAN arbeiten, als ob er lokal arbeiten würde. Er kann den LAN-Server benutzen oder eine Verbindung zum Großrechner herstellen.

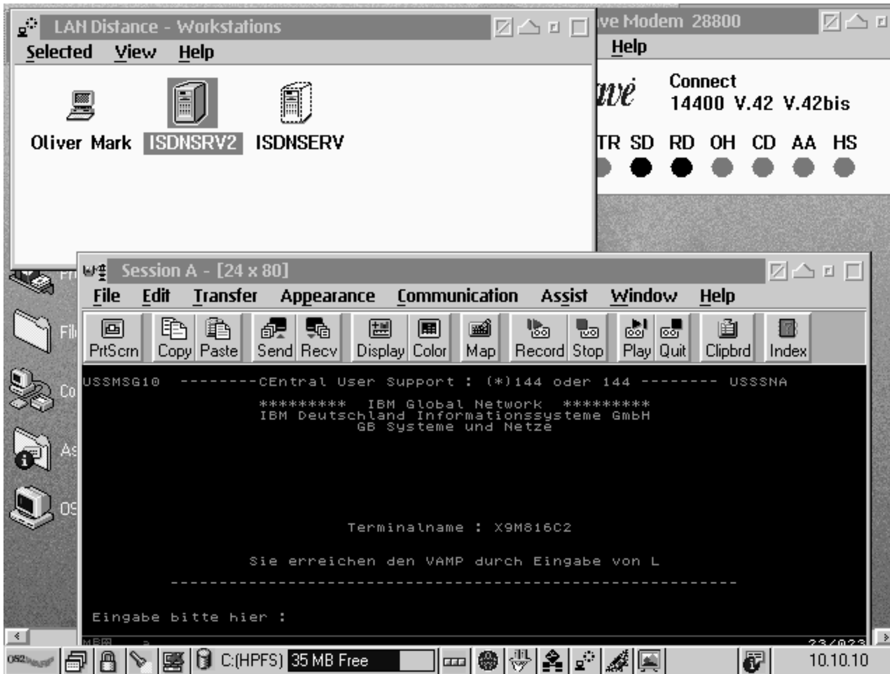


Abb. 9.87: LAN-Umgebung

9.11.4 ISDN-Unterstützung installieren

Ein Nachteil der Remote Access Services ist das Fehlen einer Standard-ISDN-Unterstützung bei der Installation. Nach der Basisinstallation finden Sie, wie eingangs schon erwähnt, lediglich die Unterstützung von analogen Modems, obwohl eine Verbindung auch über ISDN möglich ist. Voraussetzung ist hierbei natürlich, daß beide Seite ISDN unterstützen; praktisch und arbeitserleichternd ist es zudem, wenn beide Seiten die gleiche ISDN-Karte verwenden, und so zumindest Hardware-Unverträglichkeiten ausgeschlossen werden können, denn die ISDN-Unterstützung für Remote Access Services ist leider nicht trivial.

Um unter OS/2 den entsprechenden Support für ISDN-Verbindungen zu erhalten, brauchen Sie vom Hersteller der Karte zweierlei:

– OS/2-CAPI-Treiber

– ANDIS-Treiber passend zur CAPI für Remote Access Services

Die unter OS/2 beliebteste ISDN-Karte ist zum Zeitpunkt der Drucklegung die Teles S0.16. Dies hat zum einen Performancegründe, sie ist eine der schnellsten und stabilsten, zum anderen liefert Teles, als einer der wenigen, sämtliche Treiber für alle gängigen Betriebssysteme. Beim Kauf sind zwar üblicherweise nur die Windows-3.x-Treiber im Lieferumfang enthalten, aber über eine Mailbox kann man die restlichen Treiber einfach beziehen.

Installieren Sie zunächst die CAPI-Treiber:



Abb. 9.88: OS/2-CAPI-Installation

Hierzu müssen Sie wissen, welches IDSN Sie verwenden. Dies haben Sie bei der Beantragung angegeben. Entweder Euro-ISDN (DSS1) oder nationales ISDN (1TR6). Punkt zu Punkt-Verbindungen sind in seltenen Fällen notwendig, normalerweise wird das Punkt zu Mehrpunkt-Protokoll verwendet.

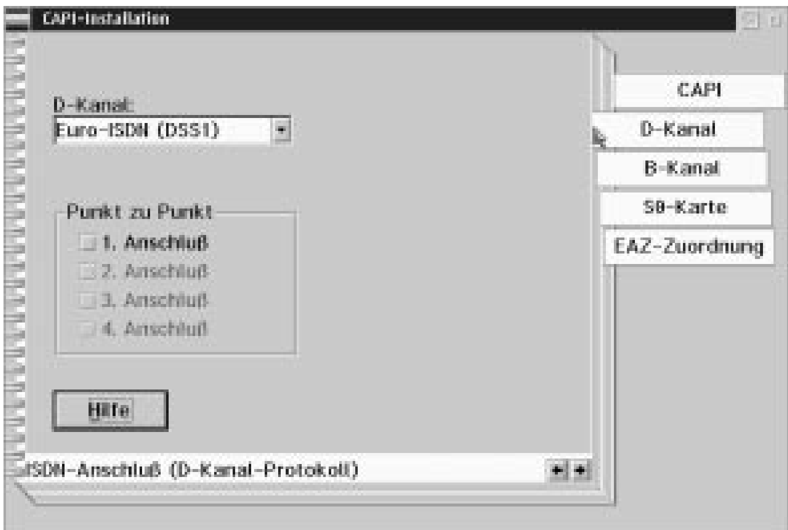


Abb. 9.89: Auswahl ISDN-Typ

Wichtig ist natürlich die Einstellungen der CAPI entsprechend der Konfiguration der ISDN-Karte vorzunehmen. Keinesfalls sollten Sie die ISDN-Karte auf einen Interrupt legen, der bereits von einer anderen Karte benutzt wird (Netzwerkkarten bevorzugen gleichfalls den IRQ 5).

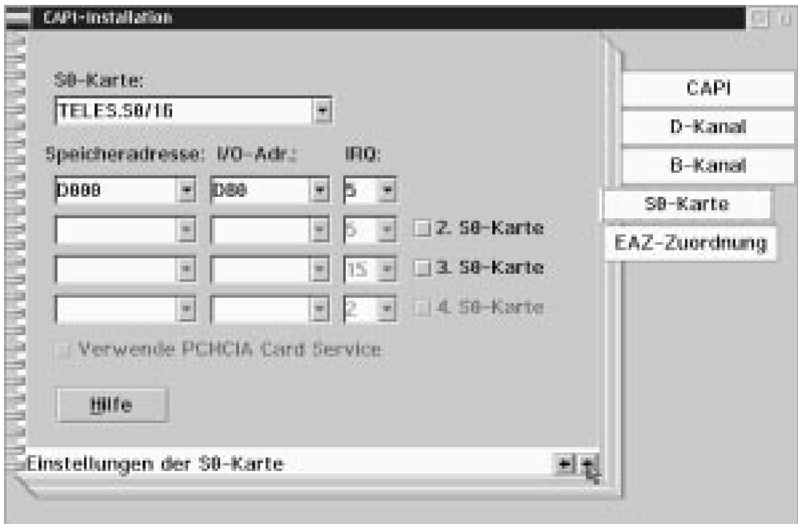


Abb. 9.90: Adreßeinstellungen ISDN-Karte

Um eine Verbindung aufbauen zu können und dem ISDN mitzuteilen, auf welcher Rufnummer er den Dienst benutzen darf und unter Umständen auf welchen Rufnummern dieser in einer Telefonanlage konfiguriert ist, müssen Sie die EAZ-Zuordnung vornehmen. Hier sollten Sie bei einem Privatanschluß mit drei Rufnummern die letzte bzw. die letzten beiden Rufnummern eintragen, vorzugsweise nicht die Hauptnummer unter der Sie üblicherweise angerufen werden.

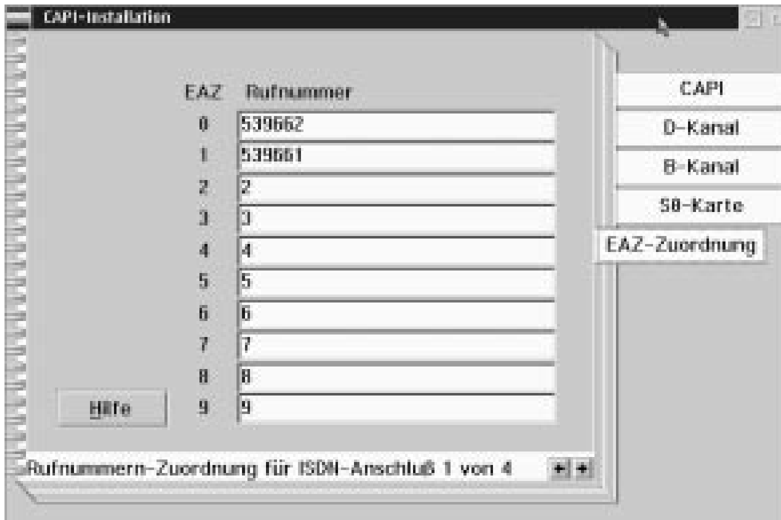


Abb. 9.91: EAZ-Zuordnung

Danach müssen Sie den Rechner neu booten, um die entsprechende CAPI-Unterstützung zu erhalten.

Jetzt können Sie die ANDIS-Treiber installieren. Das Installationsprogramm erkennt automatisch Ihre installierten Remote Access Services und die Installation der CAPI.

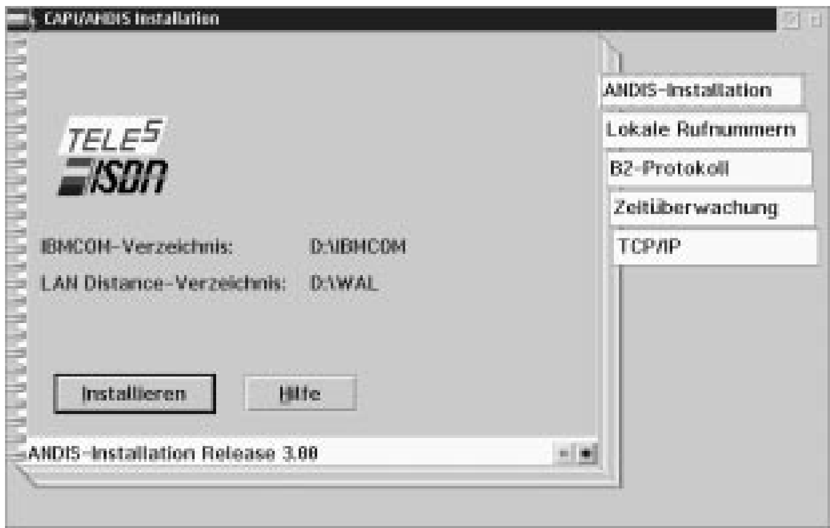


Abb. 9.92: ANDIS-Treiber-Installation

Unter der Auswahl der lokalen Rufnummern müssen Sie nun den einzelnen Kanälen die entsprechenden EAZs zuordnen. So kann Remote Access Services auf zwei Kanälen gleichzeitig zwei unterschiedliche Verbindungen zu Servern unterhalten. Dies funktioniert ausschließlich in der OS/2-Version, nicht in den Windows-Versionen. Eine Kanalbündelung auf 128 Kbit/s ist momentan noch nicht möglich, wird aber seitens der IBM angestrebt.

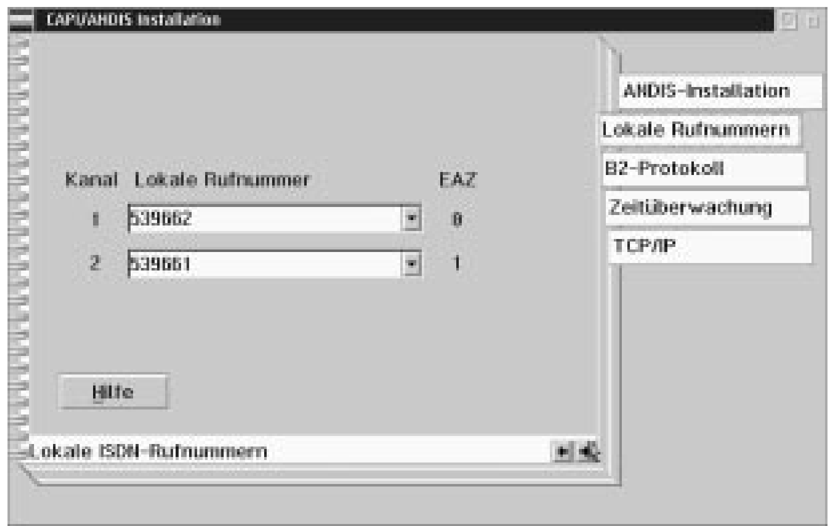


Abb. 9.93: EAZ-Zuordnung

Die Installation trägt nun alle notwendigen Treiber in die Config.Sys und die Protocol.Ini ein. Ein Neustart bewirkt die Aktivierung. Sie finden nun unter den Einträgen *Ports* und *Modem* in der Remote-Access-Services-Konfiguration einen ISDN Eintrag. Erstellen Sie unter *Verbindungen* einfach einen neuen Eintrag mit den entsprechenden ISDN-Nummern. Et voilà.

9.12 Mobile-Office-Services / Mobile File Sync

von Oliver Mark

Eine LAN-Verbindung vorausgesetzt, ist es recht einfach, mit Ressourcen eines Servers zu arbeiten. Lediglich außerhalb dieser Umgebung ergibt sich das Problem, weiterhin auf diese Ressourcen zuzugreifen. So zum Beispiel, wenn Sie mit Ihrem Notebook auf Reisen gehen. Oder aber auch die kurzzeitige Unterbrechung von Remote-Access-Verbindung oder Infrarot-Strecken. Grundsätzlich gibt es zwei Alternativen, die dieses Problem lösen können:

– Herstellen einer Verbindung über eine WAN-Strecke (zum Beispiel Telefon). Hierzu schauen Sie bitte ins Kapitel »Remote Access Services«.

– Spiegelung der LAN-Laufwerke auf den lokalen PC und bei erneuter Verbindung zum Server, Datenabgleich mit dem Serverlaufwerk. Hierzu lesen Sie bitte hier weiter.

Unter OS/2 Version 4 wird die zweite Alternative durch den Dienst »Mobile Office Service« zur Verfügung gestellt. Dieser benutzt ein »Installable File System«, nämlich das »Mobile File Sync«, kurz MFS. Hier bestimmen Sie, mit welchen LAN-Laufwerken Sie offline weiterarbeiten wollen und bei einem erneuten Logon am entsprechenden Server wieder aktualisiert werden.

Für die Installation von MFS ist der entsprechende LAN-Client Voraussetzung. Momentan werden folgende Verbindungen unterstützt: LAN Server 4.0, Warp Server, OS/2 Peer Services, Netware 4.1 und höher mit Netware Directory Services und Windows NT

Die Installation erfolgt ausschließlich auf dem Client und benötigt keine Änderungen am jeweiligen Server.

9.12.1 Installation

Standardinstallation

Die Auswahl der »Mobile Office Services« während der Erstinstallation oder innerhalb einer selektiven Nachinstallation zeigt sehr anschaulich, was sich hinter MFS verbirgt. Natürlich können Sie auch x:\CID\IMG\MFS\INSTALL.EXE aufrufen.



Abb. 9.94: Installation

Die Installation benutzt den Software-Installer der IBM. Mit der Auswahl von *Continue* kommen Sie auf die Standardpanels zur Auswahl *Config.Sys aktualisieren* und *Diskspace*. Wählen Sie *Select All*, um die Komponenten zu installieren. Bei *Diskspace* können Sie angeben, auf welches Laufwerk Sie die Installation durchführen wollen. Mit der Checkbox *Change Directories to selected drive* setzt das Installtionsprogramm automatisch alle Pfadangaben um.

Nach der Installation verlassen Sie das Eingangsbild mit *Exit*.

De-Installation

Sollten Sie das Produkt nicht mehr benötigen, können Sie mittels des Software-Installers auch die De-Installation durchführen. Dazu sollten alle MFS-Dienste gestoppt sein.

Wechseln Sie in das Verzeichnis x:\CID\IMG\MFS auf der Installations-CD oder legen Sie MFS-Diskette 1 ins Laufwerk und geben Sie *Install* ein. Danach wählen Sie *Continue* im folgenden Fenster aus. Markieren Sie den Kontrollknopf *Delete the installed product and reinstall* und wählen Sie *delete* im nächsten Fenster aus, daraufhin wird die De-Installation durchgeführt. Beenden Sie das Installationsfenster mit *Cancel* und *Exit*.

CID-Installation

Für eine Installation von zentraler Seite aus, entweder Benutzer-gesteuert (attended) oder automatisch (unattended), hat die IBM in Ihren Produkten CID-Support (Configuration, Installation, Distribution) integriert. Dies erlaubt:

- Umgeleitete Installation: Die Möglichkeit, von jedem beliebigen Laufwerk (Diskette, CD, LAN-Laufwerk) zu installieren.
- Response-File-Unterstützung: Alle einzugebenden Parameter in den Fenstern des Installationsprogrammes können mittels Parameter in einem Text-file angegeben werden.
- Kommandozeilenunterstützung: Alle Installationsparameter können auch in einer Kommandozeile eingegeben werden und müssen nicht zwangsläufig mittels grafischer Oberfläche erfolgen.
- Logfile-Unterstützung: Alle Meldungen, die normalerweise als grafisches Fenster oder als Text ausgegeben werden, können in einem (oder mehreren) Files gespeichert werden.
- Standard Return-Codes: Alle Installationsprogramme melden mit gleichem Return-Code die verschiedenen Ergebnisse einer Installation.

00 00	Installation erfolgreich
FE 04	Warnings sind aufgetreten
FE 12	Errors sind aufgetreten
16 00	Inkorrekte Aufruf des

Die Installation von MFS wird per CID folgendermaßen durchgeführt:

```
install
  /X   unattended Installation
  /A:I   Aktion ist Installation
     I: Installation
     R: Restore
     U: Update
     D: Delete
  /C:x:\cid\img\mfs\mars.icf
     Pfad zum Catalogfile (MFS-spezifisch)
  /G:x:\cid\rsp\mfs.rsp
     Pfad zum Resonsefile mit notwendigen Parametern
  /L1:x:\cid\log\mfs.l1
     Pfad zum Error-Log-File
  /L2:x:\cid\log\mfs.l2
     Pfad zum History-Log-File
  /O:DRIVE   Typ des Ursprungslaufwerks
      DRIVE         Laufwerk, lokal oder remote
     MVS, VM, VSE   Gastsysteme
  /S:x:\cid\img\mfs Quellaufwerk der Installation
  /T:C:             Ziellaufwerk der Installation
```

Im beschriebenen Response-File könnten folgende Einträge stehen (weitere im Kapitel Tuning):

```
file=c:\mfs
comp= Mobile File Sync Program
cfgupdate=auto
overwrite=no
savebackup=no
deletebackup=no
```

9.12.2 Benutzung von MFS

Nach der Installation von MFS finden Sie einen neuen Ordner auf der Arbeitsoberfläche



Abb. 9.95: Ordner MFS

Wichtiges Merkmal ist der wechselnde Text unter dem Symbol *MFS Status*, der die Zustände von MFS repräseniert.

Zustände von MFS

MFS kann, je nach Einsatz im LAN oder ohne LAN-Verbindung, verschiedene Zustände haben, die über das Symbol *MFS Status* angezeigt werden.

MFS-Status	Bedeutung
NONE	Keine Verbindung, Programm nicht gestartet.
BYE	Unerwartete Beendigung von MFS
CONN	Client und Server sind verbunden, MFS ist aktiv
DISC	
LAN	Client und Server sind getrennt, MFS ist aktiv
EXIT	MFS wurde ordnungsgemäß beendet
MATCH	Ein Fehler bei der Laufwerkszuordnung ist aufgetreten
REINT	Der Datenabgleich zwischen Client und Server ist aktiv
START	MFS wurde gestartet, Netzwerkverbindung muß eingegeben werden

Benutzung

Um MFS zu starten, klicken Sie das Symbol *MFS Start* doppelt an. Sollte der LAN- oder Peer-Client nicht aktiv sein, wird angefragt, ob dieser automatisch gestartet werden soll. MFS startet den LAN-Client mit dem Aufruf von der LOGON-Prozedur, so daß Sie bei einem installierten Peer-Client automatisch den Peer-Logon bekommen, bei einem installierten LAN-Requester automatisch einen LAN-Logon. Nachdem die LAN-Verbindung hergestellt ist, kann MFS benutzt werden.

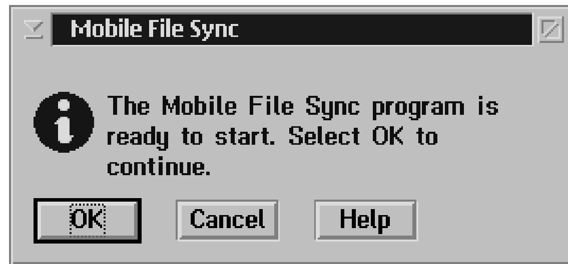


Abb. 9.96: Ready to start

Nun erscheint ein Fenster, in dem Sie angeben können, welches Laufwerk Sie mittels MFS spiegeln wollen. Haben Sie in Ihrem LOGON-Profile bereits eine Alias-Zuordnung, brauchen Sie hier lediglich den Laufwerksbuchstaben des LAN-Laufwerkes einzugeben. Falls nicht, müssen Sie den UNC1-Namen eingeben, wie zum Beispiel \\server\aliasname.

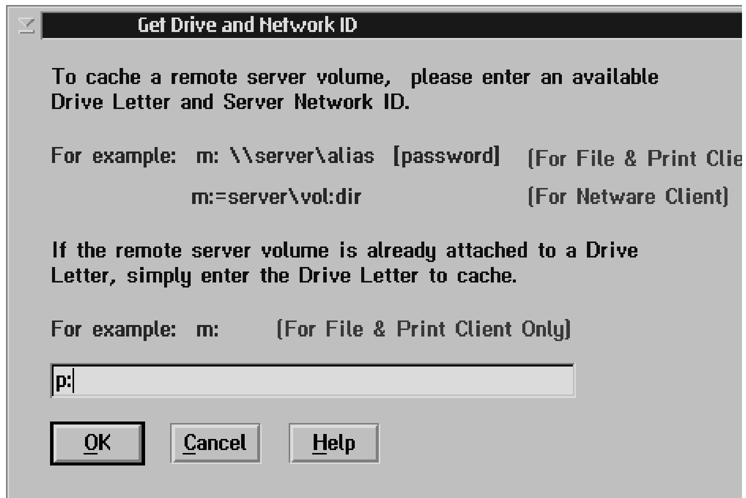


Abb. 9.97: Laufwerkszuordnung MFS

Die erfolgreiche Zuordnung des Laufwerkes wird nun sichtbar.



Abb. 9.98: Erfolgreiche MFS-Zuordnung

Der Status des MFS-Symbols ändert sich auf Connected.

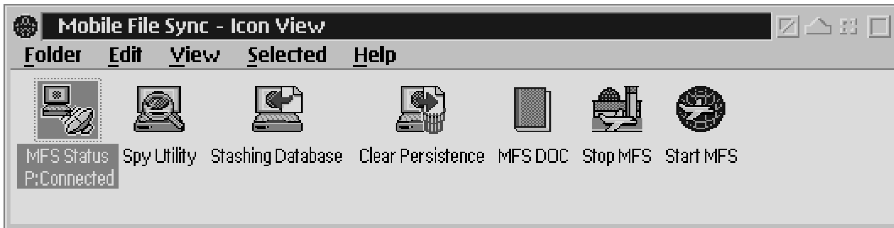


Abb. 9.99: Connected MFS

Sämtliche Daten, die nun im LAN-Laufwerk benutzt werden, werden gleichzeitig auf die lokale Platte, auf der MFS installiert ist, gespiegelt. Die Funktionen, die hierfür benutzt werden, sind:

– Implizites Caching: Alle Daten, die vom Benutzer während einer aktiven LAN-Verbindung auf dem LAN-Laufwerk aufgerufen, editiert oder benutzt werden, werden gleichzeitig auf der lokalen Platten gespiegelt. Nicht genutzte Dateien werden nicht gespiegelt, außer dies wird explizit angegeben, siehe hierzu »Stashing Database«.

– Directory Caching: Jedes aufgerufene Directory auf dem LAN-Laufwerk wird gleichfalls gespiegelt; das heißt, es werden lediglich die Directory-Einträge lokal vorgehalten, nicht die realen Dateien. Hier wird quasi ein DIR-Kommando auf die lokale Platte umgeleitet.

– File Caching: Nur die Files, die geöffnet werden, werden auch auf der lokalen Platte gespiegelt.

So können Sie nun mittels Editor Dateien öffnen, Programme ausführen, Dateien kopieren, also alle Operationen wie gewohnt auf dem LAN-Laufwerk durchführen.

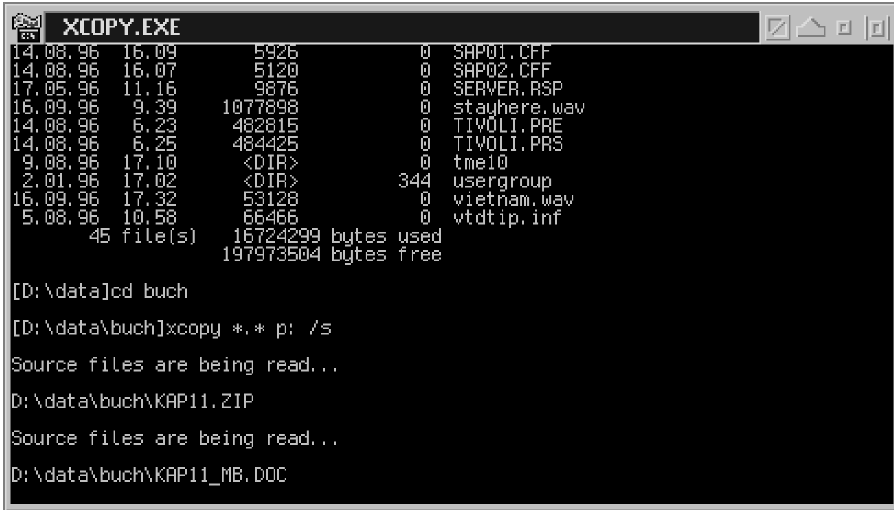


Abb. 9.100: Kopiervorgang LAN-Laufwerk

Wenn nun die LAN-Verbindung beendet wird, so sind unter dem LAN-Laufwerksbuchstaben weiterhin die benutzten Dateien vorhanden. Sie können weiterhin mit einem Editor oder einer Textverarbeitung die Dateien editieren, Programme ausführen oder Dateien löschen. Neue Dateien können jederzeit hinzugefügt werden.

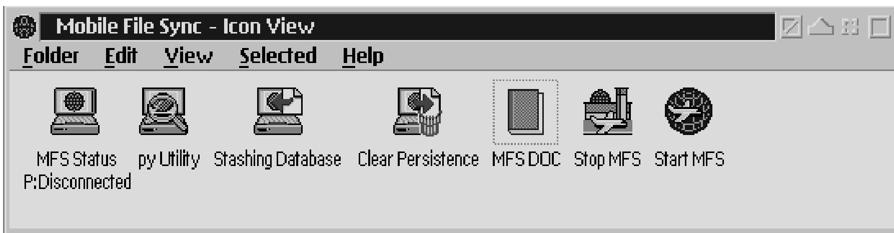


Abb. 9.101: Disconnected MFS

Wird nun die LAN-Verbindung wieder hergestellt, so versucht MFS, das gecachte Laufwerk wieder anzusprechen und eine Datenaktualisierung vorzunehmen. Wird zum Beispiel der PC in einem anderen Netzwerk angeschlossen, so kann MFS keine korrekte Verbindung herstellen und die Daten bleiben lokal. Eine Verbindung in der korrekten LAN-Umgebung startet automatisch den Reintegrationsprozess.

Alle Änderungen werden nun mit dem LAN-Laufwerk abgeglichen. Alle neuen oder geänderten Dateien des Clients werden auf dem Server aktualisiert. Sind die Daten sowohl auf dem Server als auch auf dem Client geändert worden, meldet MFS einen Konflikt, da nun zu klären ist, welches die aktuelle bzw. korrekte Version der Datei ist. Dabei unterscheidet MFS zwei Kategorien von Konflikten:

– Namenskonflikt: Wird während der Trennung von LAN-Laufwerk und Client sowohl auf dem Server als auch auf dem Client eine Datei mit gleichem Namen angelegt, so entsteht ein Namenskonflikt. Mögliche Aktionen, um diesen zu lösen, sind:

Discard/Löschen: Hierbei wird die Datei oder Directory auf dem Client gelöscht. Die Datei des Servers ist aktuelle Version.

Rename/Umbenennen: Die Datei oder Directory auf dem Client kann in einen eideutigen Namen umbenannt werden.

Replace/Ersetzen: Die Datei des Clients ersetzt die Datei auf dem Server, die Datei des Clients ist aktuelle Version.

– Datenkonflikt: Wird während der Trennung von LAN-Laufwerk und Client sowohl auf dem Server als auch auf dem Client eine bestehende Datei geändert, so entsteht ein Datenkonflikt. Mögliche Aktionen, um diesen zu lösen sind:

Discard/Löschen: Hierbei wird die Datei oder das Directory auf dem Client gelöscht. Die Datei des Servers ist die aktuelle Version.

Rename/Umbenennen: Die Datei oder das Directory auf dem Client kann in einen eindeutigen Namen umbenannt werden.

Replace/Ersetzen: Die Datei des Clients ersetzt die Datei auf dem Server, die Datei des Clients ist aktuelle die Version.

Append/Anhängen: Die komplette Datei des Clients wird an die bestehende Datei des Servers angehängt.

Edit/Editieren: Die Unterschiede der beiden Dateien werden mittels grafischer Oberfläche angezeigt, und die Dateien können editiert werden, um ein gemeinsames, gleiches File zu erzeugen.

9.12.3 Konfiguration und Tuning

Für das Tuning stehen einige Parameter zu Verfügung, die allesamt in der Config.Sys entweder per Standard eingetragen sind oder manuell hinzugefügt werden können:

– MFSCACHE=x:\MFS\CACHE	Dieser Eintrag gibt an, wo das LAN-Laufwerk lokal gespiegelt werden soll. Hierbei ist sinnvollerweise das Laufwerk mit dem größten freien Platz anzugeben.
– MFSCACHESIZE=x	Der Platzhalter x kann ein numerischer Wert zwischen 10 und 50 sein, wobei damit 10% bis 50% des freien Plattenplatzes des Cachelaufwerkes gemeint ist. Defaultangabe ist 10%, was zum Beispiel bei 400 Mbyte freiem Plattenplatz eine maximale Kapazität für die MFS-Spiegelung von 200 Mbyte bedeutet. Sind diese 200 Mbyte verbraucht, wird das Laufwerk als voll gemeldet.

– MFSMAXFILES=x — Der Platzhalter x kann einen numerischen Wert zwischen 1024 und 65536 annehmen, wobei damit die maximale Anzahl der gespiegelten Dateien gemeint ist. Sind auf dem LAN-Laufwerk sehr viele kleine Dateien, so sollte dieser Wert möglichst hoch gesetzt werden, da auch bei Erreichen dieses Grenzwertes das Laufwerk als voll gemeldet wird. Standardeintrag ist hier: 1024.

– MFSMINTIMEOUT=x — Der Platzhalter x kann einen numerischen Wert zwischen 1 und 65536 annehmen, wobei damit die Zeit in Sekunden gemeint ist, die mindestens vergangen sein muß, um einen Timeout zu generieren, das heißt, das Laufwerk als nicht mehr verbunden zu melden.

– MFSMAXTIMEOUT=x — Der Platzhalter x kann einen numerischen Wert zwischen 1 und 65536 annehmen, wobei damit die Zeit in Sekunden gemeint ist, die höchstens vergangen sein darf, um einen Timeout zu generieren, das heißt, das Laufwerk als nicht mehr verbunden zu melden.

Stashing Database

Die Verwendung von impliziten Caching ermöglicht es, Dateien, die benutzt wurden, zu spiegeln. In manchen Fällen ist es jedoch sinnvoll, auch Dateien zu spiegeln, die während der aktuellen LAN-Verbindung nicht benötigt wurden. Diese müssen dann explizit angegeben werden und in einer Datenbank definiert werden. Diese Datenbank heißt bei MFS »Stashing Database«.

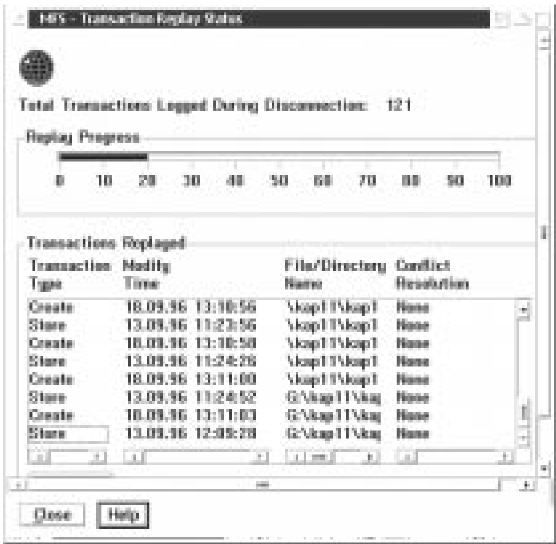


Abb. 9.102:
Stashing Database

Dort können alle Files definiert werden, die auch ohne Benutzung bei jeder LAN-Verbindung gespiegelt werden sollen. Dort können diesen Files Prioritäten mitgegeben werden, die dazu dienen, Files bei Überlauf des Caches zu löschen. Hierbei wird die niedrigste Priorität zuerst gelöscht.

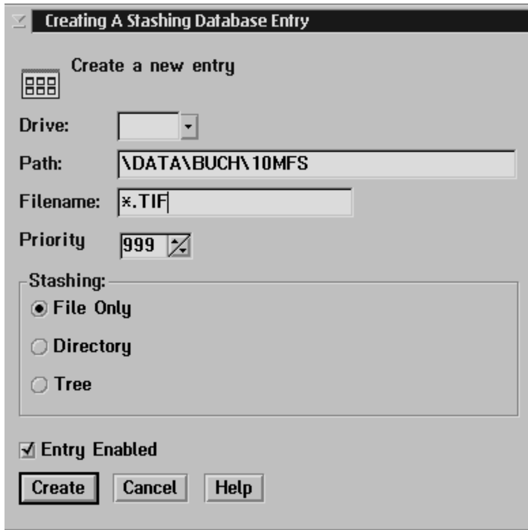


Abb. 9.103: Stashing-Database-Editor

Die angelegten File- und Directory-Listen können in Profiles abgelegt werden, um so einfach auf viele Clients verteilt werden zu können.

Über diesen Automatismus können zum Beispiel Außendienstmitarbeiter die jeweils aktuellen Informationen über Teile, Kunden, o.ä. erhalten, ohne sich jeweils darum kümmern zu müssen, woher sie diese bekommen, und was der aktuelle Stand ist. Eine grafische Anzeige dokumentiert den Fileaustausch und dessen Ende.

Innerhalb der Stashing Database können die einzelnen Files verschiedene Stati annehmen:

– *Cached*: Die Datei wurde erfolgreich gespiegelt.

– *Not Cached*: Die Datei ist nicht gespiegelt, der Eintrag wurde nach der letzten LAN-Verbindung vorgenommen.

– *Not Available*: MFS hat versucht die Datei zu spiegeln, sie konnte jedoch nicht auf dem LAN-Laufwerk gefunden werden. Der Eintrag ist entweder falsch oder kann gelöscht werden.

– *Stash Child*: Zeigt an, daß die Dateien eines Verzeichnisses mitgespiegelt werden.

– *Stash Desendants*: Zeigt an, daß die Unterverzeichnisse eines Verzeichnisses mitgespiegelt werden.

Mittels der Menüleiste des Stashing-Database-Editors können alle Aktionen durchgeführt werden:

– Table/Tabelle:

Sort/Sortieren: Sortiert die Datenbank nach Namen oder Priorität

Select all/Alle auswählen: Alle Einträge der Datenbank auswählen

Deselect all/Auswahl zurücknehmen: Alle Markierungen in der Datenbank werden gelöscht.

Read/Lesen Profile: Ein vorhandenes Profile einlesen.

Save/Speichern Profile: Die Datenbank als Profile abspeichern, um sie in einem anderen PC wiederzuverwenden.

– Entry

Copy/Kopieren: Einen neuen Datenbankeintrag erzeugen, und den ausgewählten als Vorlage benutzen. Hilfreich bei vielen ähnlichen Einträgen.

Create/Erstellen: Einen neuen Datenbankeintrag erzeugen.

Change/Ändern: Den ausgewählten Datenbankeintrag ändern.

Delete: Den ausgewählten Datenbankeintrag löschen.

– Stashing

Enable/Aktivieren: Den Stashing-Vorgang aktivieren, ohne alle eingetragenen Dateien zu spiegeln (zum Beispiel bei nicht bestehender Verbindung).

Disable/De-Aktivieren: Den Stashing-Vorgang beenden.

Stash: Den Stashing-Vorgang starten und die Spiegelung durchführen.

Spy-Utility

Dieses Utility dient dazu, aus einem laufenden Betrieb heraus ein Profile für die Stashing Database zu erstellen. Wenn Sie zum Beispiel ein Programm aufrufen, werden dadurch verschiedene andere Dateien benutzt bzw. geladen. Das Spy-Utility schreibt dies mit. Ist die Anwendung beendet, können Sie mittels des Buttons *Stash* ein Profile für die Datenbank erstellen.



Abb. 9.104: Spy Utility

Wird der Spy-Vorgang aktiviert, werden führt das Utility folgende Vorgänge aus:

– Untersuchung aller Aktivitäten auf dem gespiegelten Laufwerk.

– Anlegen von Bookends für bestimmte Anwendungen. Der Benutzer schaltet, wie in einem Buch-Kapitel, für eine bestimmte Anwendung die Teilüberwachung ein und aus. Daraus ergibt sich eine Liste von Dateien, die diese Anwendung benutzt hat.

– Alle Bookends stellen namentliche Profiles dar. Alle Spy-Ergebnisse ohne Bookend werden in die Default *stashind database* mittels des Buttons *Stash* eingetragen.

9.13 OS/2 Warp V4 als Netzwerk-Client

von Raimund Mann

Erklärtes Ziel der IBM für die neue Version von OS/2 Warp ist es, eine optimale Netzwerkintegration zu bieten, und dies unabhängig von der verwendeten Netzwerkumgebung. Auch eine Einbindung in heterogene Netzwerkumgebungen soll problemlos möglich sein. Dazu wird die Unterstützung von

– IBM LAN Server Version 3.0, 4.0 und Warp Server
– IBM DOS Lan Services Version 4.0
– IBM PC Lan Programm (PCLP)
– Microsoft Lan Manager
– Microsoft Windows für Workgroups
– Microsoft Windows NT Workstation
– Microsoft Windows NT Advanced Server

von IBM in der Dokumentation zu OS/2 Warp V4 zugesagt.

Obwohl in der Dokumentation Microsoft Windows 95 nicht explizit erwähnt wird, ist eine Peer-Verbindung auch zu einem Microsoft-Windows-95 Rechner möglich. Der Verbindungsaufbau mit Lantastic für OS/2 ist nur von einem Lantastic-Client zu einem OS/2-Peer-Rechner möglich. Umgekehrt funktioniert der Zugriff von einem OS/2-Peer-Rechner auf einen Lantastic-für-OS/2-Client jedoch nicht.

Bevor die Anbindung an die einzelnen Netzwerke besprochen wird, wird noch ein Blick auf die Installation der Netzwerkkomponenten geworfen.

9.13.1 Installation der Netzwerkkomponenten

Erkennt die Installationsroutine bei der Standardinstallation von OS/2 Warp V4 eine Netzwerkkarte im Rechner, wird nach der Installationsroutine des Basisbetriebssystems automatisch die Installationsroutine zur Installation der Netzwerkkomponenten gestartet. Wird die installierte Netzwerkkarte nicht erkannt, was während den Tests nur bei einer PCMCIA-Netzwerkkarte eines Notebooks der Fall war, kann die Installationsroutine für die Netzwerkkomponenten separat gestartet werden und die Netzwerkkomponenten nachträglich installiert werden. Auf der Installations-CD befindet sich im Verzeichnis »\ibmint« das Installationsprogramm »npinst.exe« für die Installation der Netzwerkkomponenten. Das Installationsprogramm entspricht dem, wie man es vom Warp Server her kennt. Dies hat den Vorteil, daß alle Komponenten mittels eines Installationsprogramms installiert werden können. Andererseits gibt es beispielsweise keine Möglichkeit zu entscheiden, ob als Basistreiber für die Netzwerkkarte ein ODI- oder ein NDIS-Treiber verwendet werden soll. Dies dürfte normalerweise nur dann eine Rolle spielen, wenn nur der Novell-Netware-Requester verwendet werden soll. Hier ist auf jeden Fall Handarbeit notwendig.

Die eigentliche Installation gliedert sich in zwei Komponenten. Eine Dialogbox, in der die zu installierenden Netzwerkkomponenten ausgewählt werden, und in eine Art Notizbuch, in dem die Einstellungen für die gewählten Netzwerkkomponenten getroffen werden können.

Es stehen folgende Komponenten zur Verfügung.

- LAN Requester- und Peer-Dienste: diese Option installiert den neuen IBM Requester Version 5, der eine Kombination des IBM Lan Requesters und IBM Peer darstellt. Über ihn ist der Zugriff sowohl auf Aliase von Netzwerkdomänen als auch auf von Peer-Clients (OS/2, Windows NT usw.) zur Verfügung gestellten Ressourcen möglich.
- TCP/IP-Dienste: diese Option installiert IBM TCP/IP Version 3.2
- Fernzugriffs-Client: diese Option installiert einen Lan-Distance-Client
- System-Management-Client: diese Option installiert einen System View Client in der Version 1.1
- Netware-Client: diese Option installiert den Novell Netware Requester in der Version 4.10
- MFS (Mobile Office Service): diese Option installiert eine Funktion, über die eine Dateiabgleich zwischen einem Notebook und Dateien auf einem Dateiserver möglich ist. Hierbei wird überprüft, welche Dateien auf welchem Rechner aktueller sind.

Basierend auf der getroffenen Auswahl wird anschließend eine Art Notizbuch generiert, in dem dann die Einstellungen für die einzelnen Komponenten vorgenommen werden können.

Hierbei kann man für die meisten Einstellungen die Vorgabewerte übernehmen. Anhand der Markierung vor den einzelnen Punkten auf der linken Seite des Notizbuchs ist zu erkennen, ob eine Konfiguration notwendig ist oder nicht.

Eine rote schließende spitze Klammer zeigt an, daß dieser Punkt konfiguriert werden muß.

Ein waagrechter Strich bedeutet, daß dieser Punkt mit den Vorgabewerten konfiguriert wird

Ein Häkchen bedeutet, daß dieser Punkt vom Anwender bereits konfiguriert wurde..

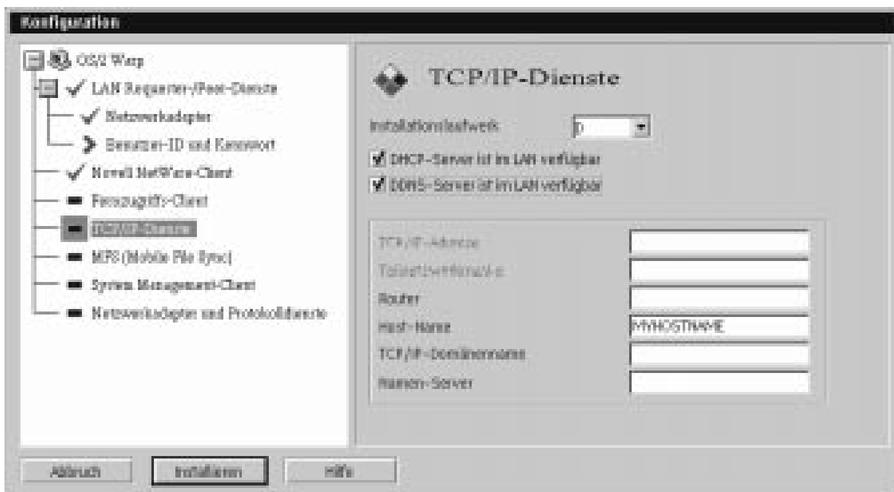


Abb. 9.105: Einstellungs-Notizbuch der Netzwerkkomponenten

Sind alle Punkte konfiguriert oder wurden zumindest die Vorgabewerte akzeptiert, werden nach einer Rückfrage im nächsten Schritt die ausgewählten Netzwerkkomponenten auf den Rechner installiert.

Nach einem Neustart des Rechners steht der Netzwerkanbindung dann nichts mehr im Weg.

Soll eine Anbindung an ein Novell-Netware-Netzwerk erfolgen, gilt es jedoch, noch die ein oder andere Einstellung vorzunehmen, bevor man die Netzwerkkomponenten installiert. Das Installationsprogramm geht nämlich davon aus, daß die Token-Ring-Topologie zum Einsatz kommt und trägt daher diesen Rahmentyp in die »protocol.ini« (im Verzeichnis »\ibmcom«) ein. Wird dagegen die Ethernet-Topologie verwendet, muß im Notizbuch für die Einstellungen noch der zu verwendende Rahmentyp angegeben werden. Doch dazu später mehr.

Der erste Punkt in der Liste der zu konfigurierenden Netzwerkkomponenten betrifft den neuen IBM Requester. Hier ist neben dem Laufwerk, auf dem die Software installiert werden soll, noch der NetBios-Name des Rechners, optional eine Beschreibung, sowie die Domäne, in der

sich der Rechner befindet, einzutragen. Hierbei ist zu beachten, daß der NetBios-Name des Rechners innerhalb der angegebenen Domäne eindeutig ist. Ansonsten kann nach der Installation der Requester nicht gestartet werden. Als Optionen kann ferner gewählt werden, ob die grafischen Lan-Administrations-Komponente installiert werden soll. Dies beinhaltet die GUI-Anwendungen und Objekte, wie sie mit dem IBM LAN Server 4.0 eingeführt wurden.

Mit der nächsten Option wird die Peer-Funktionalität installiert. Diese ist notwendig, um eine Ressource eines anderen Rechners verwenden zu können bzw. eine eigene Ressource zur Verfügung zu stellen.

Als letzte Option kann entschieden werden, ob die Domain Control Database (DCDB) auf dem Rechner gelöscht und neu initialisiert werden soll oder nicht. In der DCDB werden alle Benutzer- und Gruppendefinitionen sowie die Ressourcen verwaltet. Wird die DCDB gelöscht, wird auch die Datei »net.acc« neu angelegt, in der die Zugriffsberechtigungen der Benutzer und Gruppen hinterlegt sind.

Als erster Unterpunkt zu den Einstellungen für den IBM Requester kann der zu verwendende Netzwerkadapter angegeben werden. Es werden bis zu vier Netzwerkkarten unterstützt, die von 0 bis 3 numeriert werden.

Beim zweitem Unterpunkt ist die Benutzerkennung und das Paßwort für den Requester anzugeben. Dies wird die Administrator-ID für den Rechner, um eigene Ressourcen freigeben zu können. Soll gleichzeitig auf ein Netzwerk und auf Peer-Rechner zugegriffen werden, empfiehlt es sich, sowohl für das Netzwerk als auch für die Peer-Funktionalität die gleiche Benutzerkennung und das gleiche Paßwort zu verwenden. Andernfalls kann nach der Installation mit dem zum Lieferumfang gehörendem Network SignOn Coordinator ein Anmelden mit einer einzigen Benutzerkennung und Paßwort an unterschiedliche Rechner bzw. Netzwerke realisiert werden.

Zur Konfiguration des Novell-NetWare-Clients steht nur die Option zur Verfügung, ob ein Bindery-basierendes Netzwerk oder ein Netware-4.x-Netzwerk unter Verwendung der NDS (NetWare Directory Services) verwendet wird. Wird ein NDS-basierendes Netzwerk verwendet, kann noch der zu verwendende Kontext innerhalb der NDS angegeben werden. Ansonsten kann nur der Server angegeben werden, der standardmäßig verwendet werden soll. Die hier getroffenen Einstellungen werden in der Datei »net.cfg« im Abschnitt *Netware Requester* unter den Schlüsselwörtern *Netware Directory Services Off|On, Preferred Server* bzw. *Name Context* eingetragen. Die Datei »net.cfg« wird von der Installationsroutine im Root-Verzeichnis des Boot-Laufwerks angelegt.

In direktem Zusammenhang mit der Konfiguration des NetWare-Requesters steht die Konfiguration der Netzwerkprotokolle. Besser gesagt, die Konfiguration der NetWare-Requester-Unterstützung durch MPTS (Multi Protocol Transport Service). Hiermit wären wir auch schon bei der Konfiguration der Netzwerkkarte(n) und Protokolle.

Die Installationsroutine listet hier die erkannten Netzwerkkarten und die für die ausgewählten Netzwerkkomponenten notwendigen Protokolle auf. Über die entsprechenden Schaltflächen auf der rechten Seite lassen sich weitere Netzwerkkarten und Netzwerkprotokolle hinzufügen beziehungsweise löschen.

Wurde die installierte Netzwerkkarte nicht richtig erkannt oder soll eine weitere Netzwerkkarte eingetragen werden, kann über die Schaltfläche *Adapter ändern* die Dialogbox zur Auswahl der Netzwerkkarte geöffnet werden. Hier werden dann alle direkt von OS/2 Warp V4 unterstützten Netzwerkkarten aufgelistet. Befindet sich die installierte Netzwerkkarte nicht in der Liste der direkt unterstützten Netzwerkkarten, kann über die Schaltfläche

Anderer Adapter die Dialogbox geöffnet werden, in der der Pfad zu den vom Hersteller der Karte mitgelieferten Dateien angegeben werden kann. Wird die entsprechende Nif-Datei im angegebenen Pfad gefunden, werden die Dateien mit den Endungen »*.nif«- und »*.os2« (eigentliche Kartentreiber) in das Verzeichnis »\ibmcom\macs« kopiert. Anschließend steht die neue Netzwerkkarte in der Auswahl der Netzwerkkarten zur Verfügung.

Ähnlich verhält es sich mit den Netzwerkprotokollen. Wollen Sie ein Protokoll verwenden, das nicht zum Lieferumfang von OS/2 Warp gehört, und haben Sie die Dateien für eine entsprechende Unterstützung eingebunden, können Sie das neue Protokoll über die Schaltfläche *Protokoll hinzufügen* einbinden.

Die folgenden Protokolle werden mit OS/2 Warp mitgeliefert, womit auch die meisten Netzwerkumgebungen abgedeckt sein dürften:

- IBM IEEE 802.2
- IBM OS/2 NetBios
- IBM OS/2 NetBios über TCP/IP
- IBM TCP/IP
- Unterstützung für IBM NetWare Requester
- NetWare-NetBios-Emulation über IPX

Viel wichtiger ist, daß auf dieser Seite des Notizbuchs die Einstellungen für die Netzwerkkarten und Protokolle getroffen werden können. Bei der Netzwerkkarte wäre hier auf jeden Fall der verwendete Interrupt sowie die Portadresse einzutragen und zu überprüfen, sofern dies für die Karte notwendig ist. Einige Karten wie beispielsweise die 3C5X9 von 3COM benötigen diese Angaben nicht. Für die NE2000 jedoch müssen diese Angaben gemacht werden. Hierzu ist die Netzwerkkarte auszuwählen und über die Schaltfläche *Einstellungen* den Dialog zur Konfiguration zu öffnen. Hierbei wird von der Installationsroutine die zur Netzwerkkarte gehörende »*.nif« Datei aus dem Verzeichnis »\ibmcom\macs« ausgelesen und die darin definierten Schlüsselwörter zur Konfiguration angeboten.

In dieser Dialogbox wird dann der Beschreibungstext sowie die Bereichsangabe zu dem ausgewählten Schlüsselwort angezeigt. Diese Informationen werden ebenfalls der »*.nif« Datei entnommen.

Um einen Wert zu ändern, muß leider etwas umständlich jedes Schlüsselwort einzeln angeklickt werden und über die Schaltfläche *Ändern* die Dialogbox zum Aktualisieren des Wertes aufgerufen werden.

Um die Einstellungen für die einzelnen Protokolle zu ändern, ist die Vorgehensweise identisch. Im Normalfall sind hier jedoch keine Änderungen notwendig mit Ausnahme der Protokolle NetBios über TCP/IP und NetWare-Requester-Unterstützung. Für die NetWare-Requester-Unterstützung ist in den Einstellungen auf jeden Fall der zu verwendende Rahmentyp anzugeben. Als Vorgabe wird der Rahmentyp TOKEN-RING verwendet. Wird in der Netzwerkumgebung jedoch eine ETHERNET-Topologie verwendet, ist bei dem korrektem Rahmentyp der entsprechende Wert auf *yes* zu setzen. Bei Novell NetWare ab Version 3.12 ist Rahmentyp ETHERNET_802.2 der Standard-Rahmentyp. Bei allen vorhergehenden Versionen von NetWare ist es der Rahmentyp ETHERNET_802.3. Welcher Rahmentyp in der aktuellen Umgebung verwendet wird, ist vom jeweiligem Netzwerkadministrator zu erfahren. Bei der zur Verfügung stehenden Version von OS/2 Warp 4 wurde die Einstellung für den Rahmentyp jedoch nicht in die »protocol.ini« zurückgeschrieben, so daß nach dem Booten der Fehler »REQ0815 ...« auftrat. In diesem Fall kann man die Einträge in der »protocol.ini« (im Verzeichnis »\ibmcom«) entweder per Hand ändern oder über MPTS (Multi Protocol Transport Service) die Konfiguration erneut vornehmen.

Die Konfiguration der TCP/IP-Komponente hängt davon ab, ob innerhalb des TCP/IP-Netzwerkes ein DHCP- (Dynamic Host Configuration Protocol) Server, eventuell in Kombination mit einem DDNS (Dynamic Domain Name Server), verwendet wird.

Wird kein DHCP-Server verwendet, ist die TCP/IP-Adresse des Rechners sowie die Subnet Mask anzugeben.

Zusätzlich kann die Routeradresse, der Hostname des Rechners (wird in der »config.sys« unter *SET Hostname* = eingetragen), der Domäne-Name des TCP/IP-Netzwerkes und die TCP/IP-Adresse des Name-Servers angegeben werden. Die Routeradresse ist nur dann erforderlich, wenn auf andere Netzwerksegmente als das eigene zugegriffen werden soll. Dies dürfte jedoch bei den meisten TCP/IP-Netzwerkinstallationen der Fall sein, so daß die Routeradresse eingetragen werden sollte. Diese Adresse kann beim Netzwerkadministrator in Erfahrung gebracht werden.

Befindet sich im Netzwerk ein DHCP-Server, ist auf der Client-Seite keine weitere Konfiguration erforderlich als die entsprechenden Wahlfelder zu markieren. Den Rest übernehmen dann beim Booten die DHCP- und DDNS-Server.

Beim Systemstart eines als DHP-Client konfigurierten Rechners sendet dieser einen sogenannten Broadcast ins Netzwerk und wartet auf die Antwort eines DHCP-Servers. Der erste DHCP-Server, der antwortet, wird dann verwendet, um eine gültige TCP/IP-Adresse zu erfragen die vom Client dann verwendet wird. Die zur Verfügung gestellte Adresse kann von dem Client für die sogenannte Lease-Dauer verwendet werden. Diese Lease-Dauer (Zeitraum, für den die Adresse einem Client zur Verfügung gestellt wird) kann auf dem DHCP-Server konfiguriert werden. Hat der Client auf diesem Weg eine TCP/IP-Adresse erhalten, sendet er anschließend, falls konfiguriert, diese Adresse in Verbindung mit dem NetBios-Namen des Rechners an einen DDNS, der diese Information dynamisch in seine »Hosttabelle« einträgt (siehe auch »OS/2 Warp V4 als DHCP-Client«). Somit kann der im Netz jetzt neu verfügbare Rechner sowohl über seinen NetBios-Namen als auch direkt über die TCP/IP-Adresse angesprochen werden.

Optional kann auch hier wieder die Routeradresse, der Hostname des Rechners, der Domänen-Name des TCP/IP-Netzwerkes und die TCP/IP-Adresse eines Name-Servers angegeben werden.

Die Konfiguration von NetBios über TCP/IP gliedert sich in 3 Gruppen.

– Konfiguration des NetBios über TCP/IP-Treiber

– Konfiguration der Namens-Liste (Host-Datei)

– Konfiguration der Broadcast-Liste

Im Konfigurationsdialog für NetBios über TCP/IP können die Informationen bezüglich des Node-Typs, die Adresse des Name-Servers (plus Backup-Server) sowie die weiteren Konfigurationsparameter der NetBios-Schnittstelle so wie sie auch für das NetBEUI-Protokoll verwendet werden.

Über den zweiten Punkt kann eine statische Hostdatei »rfcnames.lst« im Verzeichnis »\ibmcom« angelegt werden. Diese Liste wird durchsucht, wenn ein NetBios-Name im Netzwerk nicht gefunden werden konnte.

Der letzte Punkt schließlich dient dazu, bestimmte Rechner in die Broadcast-Liste aufzunehmen. Hier kann entweder ein NetBios-Name oder die TCP/IP-Adresse eingegeben werden. Diese Information wird in der Datei »rfcbcst.lst« ebenfalls im Verzeichnis »\ibmcom« gespeichert.

Wurden alle Angaben zu den zu installierenden Komponenten gemacht und die Dateien auf dem Rechner installiert, steht nach einem Neustart des Rechners der Netzwerkintegration nichts mehr im Weg. Im Ordner *Verbindungen* werden dann alle zur Verfügung stehenden Netzwerkkomponenten angezeigt.

Treten beim Booten jedoch Fehler auf oder kann eine Verbindung zu dem Netzwerk nicht aufgebaut werden, empfiehlt es sich, im Verzeichnis »\ibmcom« die Datei »lantran.log« zu überprüfen. In diese Datei werden beim Booten des Rechners alle Informationen bezüglich der Netzwerkkomponenten eingetragen. Dazu zählen unter anderen

– die Adresse der Netzwerkkarte

– welche Protokolle geladen wurden

– welche Bindungen erfolgreich durchgeführt wurden bzw.

– welche Bindungen nicht möglich waren

– wieviele Ressourcen für die Protokolle zur Verfügung stehen

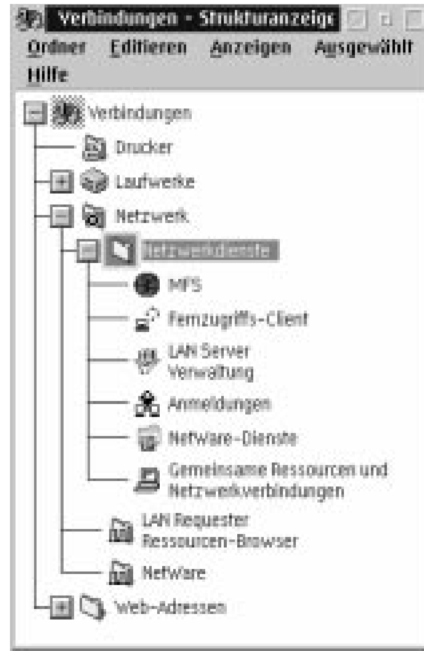


Abb. 9. 106: Der Ordner *Verbindungen* mit Netzwerkkomponenten

Anhand dieser Informationen sollte es möglich sein, den Fehler einzukreisen und durch eine entsprechende Konfiguration unter Verwendung des MPTS zu beheben.

Die Datei »lantran.log« wird bei jedem Neustart des Rechners neu geschrieben. Alte Informationen gehen dabei verloren.

Noch ein Tip zum Schluß: Wurde aus irgendeinem Grund der Requester nicht über das Installationsprogramm deinstalliert, sondern manuell gelöscht, erkennt die Installationsroutine bei einer versuchten Neuinstallation immer noch einen installierten Requester. Dies kann unter Umständen zu Problemen bei der Installation führen. Um hier Probleme zu vermeiden, ist einfach die Datei »ibmlvl.ini« aus dem Root-Verzeichnis zu löschen. Anhand dieser Datei »erkennt« die Installationsroutine, ob ein installierter Requester vorliegt oder nicht. Löscht man diese Datei, wird der Requester komplett neu installiert, auch wenn man die physikalischen Dateien nicht gelöscht hat.

9.13.2 OS/2 Warp V4 als IBM-LAN-Server-Client

Da es sich beim Requester von OS/2 Warp V4 um eine Kombination des LAN Server Requesters Version 4 und IBM Peer für OS/2 handelt, ist die Einbindung in ein IBM-LAN-Server-Netzwerk denkbar unkompliziert. Unterstützt werden dabei IBM LAN Server Version 3.x und höher. Ebenso kann mittels des neuen Requesters auf andere Requester (DOS-/Windows- oder

OS/2-Requester der Domäne) mit installierten Peer-Diensten zugegriffen werden. Allerdings erscheinen diese Rechner nicht im LAN Requester Ressourcen-Browser. Auch bei der Definition einer Verbindung erscheinen solche Computer und deren Ressourcen nicht in der Auswahlliste. Sie können jedoch explizit unter Verwendung des UNC- (Universal Naming Convention) Namens angegeben werden, um den Zugriff darauf zu ermöglichen.

Im Ordner *Verbindungen* befinden sich die zusätzlichen Icons für den LAN Requester Ressourcen-Browser sowie die Netzwerkdienste. Wird die Baumstruktur unter *Netzwerkdienste* durch Mausklick auf das Pluszeichen geöffnet, erscheint der Ordner *Anmeldungen*. Ein Doppelklick auf diesen öffnet den Ordner *Anmeldungen*, in dem sich alle Dienstprogramme für die Netzwerkintegration und Peer OS/2 befinden.

Wurden die Protokolle einwandfrei geladen und an die Karte gebunden, kann der Requester nun entweder über die Befehlszeile mittels »NET START REQ[UESTER]« (die Peer-Dienste werden automatisch mitgestartet, es sei denn, man hat dies über die »ibmlan.ini« deaktiviert) oder über das Icon *Starten LAN Requester Dienste* gestartet werden. Anschließend ist der Zugriff auf externe Ressourcen eines LAN-Server-Netzwerks möglich. Wichtig ist, daß zum Anmelden an einer LAN-Server-Domäne auch das Icon *LAN Server Anmeldung* und nicht *LAN Requester-Dienste Anmeldung bei Datenstation* verwendet wird. Der Unterschied der beiden Progammaufrufe ist die Parameterübergabe. Beim Anmeldedialog für *LAN Server Anmeldung* wird als Parameter »/V:DOMAIN /D« übergeben, was nichts anderes bedeutet, als daß die Validierung (/V steht für Verification) in der Domäne vorzunehmen ist. Im Gegensatz dazu hat der Programmaufruf der LAN-Requester-Dienste den Parameter »/V:LOCAL«.

Bei allen anderen Anmeldedialogen wird die eingegebene Benutzerkennung und das Paßwort für die lokale Benutzerdatenbank validiert, da dies die Vorgabe ist, die über die »ibmlan.ini« (im Verzeichnis »\ibmlan«) getroffen wird. Dies bedeutet, daß alle Benutzerkennungen von Benutzern dieses Rechners auch lokal definiert sein müßten. Da dies einen beträchtlichen Aufwand an Administration bedeutet, kann man in der »ibmlan.ini« die Überprüfung der Benutzerkennung auf eine Validierung in der Domäne umstellen. Hierzu ist die »ibmlan.ini« mit einem beliebigem Editor wie zum Beispiel dem EPM oder E zu laden und in der Sektion »[requester]« unter dem Schlüsselwort »wrkheuristics =« das 37. Bit auf 2 zu setzen. Damit wird standardmäßig die Validierung in der Domäne vorgenommen. Eine Validierung anhand der lokalen Benutzerdatenbank »net.acc« hätte zudem noch den Nachteil, daß Anmeldezuordnungen, die in der Domäne definiert wurden, nicht durchgeführt werden. Alternativ kann man über die »ibmlan.ini« auch noch die weiterleitende Anmeldung aktivieren. In diesem Fall würde zuerst überprüft, ob der eingegebene Benutzername und das Paßwort eine gültige Kombination auf dem lokalen Rechner darstellt. Ist dies nicht der Fall, wird bei der weiterleitenden Anmeldung der Benutzername und das Kennwort an die Domäne zur Validierung weitergeleitet. Stellt der Benutzername und das Kennwort innerhalb der Domäne eine gültige Kombination dar (existiert ein entsprechendes Benutzerkonto), so wird der Zugriff auf die Ressourcen des Rechners gewährt. Um die »Forwarded Authentification« zu aktivieren, ist in der »ibmlan.ini« in der Peer-Sektion der Wert von *forwardauth=* auf »yes« zu setzen.

Hat man sich lokal angemeldet, ist eine erneute Anmeldung an der Domäne nicht erforderlich, sofern ein äquivalentes Benutzerkonto innerhalb der Domäne existiert. Die Benutzerkennung

und das Paßwort wird beim ersten Zugriff auf eine Domänen-Ressource an die Domäne zur Validierung übermittelt und dadurch der Zugriff ohne erneute Anmeldung ermöglicht.

Für einen reinen Netzwerk-Client, der nur innerhalb einer LAN-Server-Domäne ohne eigene Ressourcenfreigabe agiert, kann das automatische Starten der Peer-Services unterbunden werden. Beim Schlüsselwort *wrkservices =* ist dazu in der Requester-Sektion der Service *Peer* zu löschen. Damit diese Änderungen wirksam werden, ist der Requester zu stoppen und erneut zu starten.

Hat man sich an einer Domäne angemeldet, stehen im LAN Requester Ressourcen-Browser alle Server dieser Domäne als Ordner zur Verfügung. Zusätzlich wird noch ein Icon *Aliase für Domäne xyz* angezeigt. Dieses Objekt beinhaltet alle definierten Aliasnamen für die Domäne xyz. Für jeden Server innerhalb der Domäne, der im Browser aufgelistet wird, kann nun über das Kontextmenü (rechte Maustaste) auf einen anderen Server zugegriffen werden sowie der Zugriff und die Freigabe auf das *Serverobjekt* auf der lokalen Maschine verwaltet werden. Das darf jedoch nicht mit der Freigabe einer Ressource auf dem entsprechenden Server verwechselt werden!

Öffnet man nun einen dieser Server-Ordner, sind alle darin enthaltenen Aliase und Freigaben, die auf diesem Server definiert sind, sichtbar. Öffnet man dagegen den Ordner »Aliase der Domäne xyz«, sind nur die in der Domäne definierten Aliase, unabhängig von dem physikalischen Server, sichtbar. Über das Kontextmenü eines Aliases oder einer Freigabe kann diese nun einem Laufwerksbuchstaben (bei Verzeichnisfreigaben), einer parallelen oder seriellen Schnittstelle zugewiesen werden oder die Zuordnung aufgehoben werden, falls bereits eine Zuordnung besteht (z.B. durch eine Anmeldezuordnung). Auf diese Weise ist es sehr einfach möglich, in grafischer Form Netzwerkverbindungen zu erstellen und zu lösen. Diese sind jedoch nur temporär und gehen beim nächsten Systemstart verloren, es sei denn, man hat in der Datei »config.sys« unter *SET AUTOSTART=* den Eintrag »CONNECTIONS«. Dieser bewirkt, daß bei einem Systemabschluß alle Netzwerkverbindungen vom System »gemerkt« und bei einem Neustart wieder aufgebaut werden.

Öffnet man die *Einstellungen* einer Freigabe (Alias oder Netzname) über das Kontextmenü, enthält das Notizbuch dieser Freigabe eine Seite *Netzwerkstatus*, über die alle Informationen zu der Freigabe eingesehen werden können. Hierzu gehören:

– die Netzwerkumgebung (LS oder NW)

– der Server, auf dem sich die Freigabe befindet

– der Name der Ressource auf dem Server

– die Beschreibung

– der Status der Freigabe

– die Bezeichnung der zugewiesenen lokalen Ressource oder »Keins«, falls keine Zuordnung vorliegt.

Um immer eine konsistente Netzwerkumgebung vorzufinden, verwendet man innerhalb einer Domäne normalerweise Anmeldezuordnungen, die in der Regel vom Administrator eingetra-

gen, jedoch vom Benutzer geändert werden können. Ist man an einer Domäne angemeldet, kann man die eigenen Anmeldezuordnungen über das Notizbuch *Netzwerkbenutzereintrag* im Ordner *Anmeldungen* selbst verwalten. Dieses Notizbuch dient dazu, für den aktuellen Benutzer die Beschreibung, das Paßwort sowie die Anmeldezuordnungen und allgemeinen Anwendungen zu verwalten. Die Seiten *Anmeldezuordnungen* und *Anwendungen* stehen jedoch nur zur Verfügung, wenn die Anmeldung an einer Domäne erfolgte. Ebenso ist das Wahlfeld *Domänenkennwort ändern* nur verfügbar, wenn eine Domänenanmeldung vorliegt. Andernfalls kann nur das lokale Paßwort geändert werden.

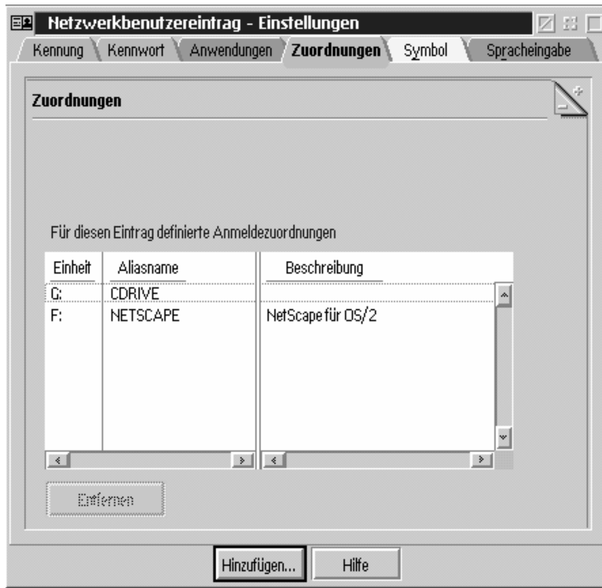


Abb. 9.107: Überwachen der Anmeldezuordnungen über Netzwerkbenutzereintrag

Klickt man auf den Tab *Zuordnungen*, erscheint eine Liste aller bereits definierten Anmeldezuordnungen. Über die Schaltfläche *Hinzufügen* kann nun eine neue Anmeldezuordnung erstellt werden. Um eine Anmeldezuordnung zu definieren, ist zuerst die Art der Ressource auszuwählen, für die eine Anmeldezuordnung erstellt werden soll. Durch Auswahl eines der drei Radiobuttons wird in dem Listenfeld sofort die Anzeige aktualisiert und alle in der Domäne verfügbaren Alias-Freigaben angezeigt (für über Netznamen freigegebene Ressourcen muß eine Netzwerkverbindungsdefinition erstellt werden). Anschließend kann der gewünschte Alias gewählt und die lokale Ressource angegeben werden (zu verwendender Laufwerksbuchstabe usw.). Mit OK wird die Dialogbox wieder verlassen, um die neu erstellte Anmeldezuordnung zu sichern. Nun erscheint die neu erstellte Anmeldezuordnung in der Liste der bereits definierten Anmeldezuordnungen und wird bei der nächsten Anmeldung berücksichtigt. Um eine neu definierte Anmeldezuordnung sofort verwenden zu können, muß entweder über den Befehl »NET USE« die Verbindung manuell erstellt werden oder der Benutzer muß sich ab- und anschließend erneut anmelden.

Soll eine Anmeldezuordnung gelöscht werden, ist diese in der Liste der Anmeldezuordnungen auszuwählen und anschließend über die Schaltfläche *Entfernen* zu löschen.

Die Vorgehensweise für allgemeine Anwendungen (Public Applications) ist identisch zu den Anmeldezuordnungen. Auch hier ist zuerst der Tab *Anwendungen* zu wählen, um eine Liste aller in der Domäne definierten allgemeinen Anwendungen anzuzeigen. Anschließend kann über die Schaltfläche *Hinzufügen* eine neue Anwendung in die Liste der allgemeinen Anwendungen aufgenommen werden.

Wurde für den Anmeldevorgang die lokale Validierung gewählt (die Vorgabe), muß man auf Anmeldezuordnungen verzichten. Der neue Requester bietet jedoch auch für diesen Fall eine Möglichkeit, automatisch bestimmte Netzwerkverbindungen wieder aufzubauen. In diesem Fall muß man sogenannten Verbindungensdefinitionen erstellen. Öffnet man das Notizbuch *Gemeinsame Ressourcen und Netzwerkverbindungen* im Ordner *Netzwerkdienste*, kann man auf der ersten Seite den aktuellen NetBios-Namen des lokalen Rechners sehen sowie dessen Beschreibung und die Standarddomäne angeben. Ferner ist noch ein Wahlfeld *LAN-Dienste beim Systemstart starten* vorhanden, das bei Aktivierung eine Referenz des Objektes *Starten LAN Requester-Dienste* im Systemstart-Ordner generiert.



Abb. 9.108: Notizbuch zum Definieren von Verbindungen und Freigaben

Auf der Seite *Verbindungen* kann man alle bereits definierten Verbindungen einsehen. Um eine neue Verbindung zu definieren, ist über die Schaltfläche *Verbindung herstellen* der Dialog für die Definition der Verbindung zu öffnen. In diesem Dialog ist anzugeben, wo sich die zu verbindende Ressource befindet. Zur Wahl stehen entweder eine Domänen-Ressource oder eine (Peer -) Server-Ressource. Hier ist der Name des (Peer –) Servers oder der Name der

Domäne anzugeben. Bei den Domänen werden alle in der »ibmlan.ini« unter *othdomain=* eingetragenen sowie die Standarddomäne aufgelistet. Aus der Liste der auf dem gewählten Server oder Domäne verfügbaren Ressourcen ist dann die gewünschte Ressource und die lokale Einheit (Laufwerksbuchstabe, paralleler oder serieller Anschluß), der die Ressource zugeordnet werden soll, auszuwählen. Hier erscheinen in der Liste nicht nur die Alias-Freigaben der Domäne, sondern auch alle Freigaben über Netznamen (mittels NET SHARE). Für eine Verbindungsdefinition wird nämlich immer die vollständige UNC (Universal Naming Convention) zu der betreffenden Ressource verwendet. Bei den lokalen Einheiten werden nur die noch zur Verfügung stehenden Einheiten angezeigt, wobei die Vorgabe die jeweils erste lokal verfügbare Einheit ist.

Für die Definition der Verbindung kann nun noch angegeben werden, ob diese Verbindung nach dem Anmelden automatisch wieder aufgebaut werden soll (eine Pseudo-Anmeldezuordnung). Dies ist die Vorgabe für neu erstellte Verbindungen. Als zweite Option wird die Möglichkeit geboten, vor dem Verbindungsaufbau ein Paßwort anzugeben, das für den Verbindungsaufbau zum Server oder der Domäne verwendet werden soll. Dies ist insbesondere bei Peer-Verbindungen hilfreich, wo unter Umständen die Passwörter auf den einzelnen Peer-Servern nicht abgeglichen sind.

Wird der Dialog zur Definition einer Verbindung mit OK verlassen, erscheint diese neue Verbindungsdefinition automatisch in der Liste der Verbindungen. Wird der Dialog zur Definition von Verbindungen mit OK verlassen, so steht die neue Verbindung – im Gegensatz zur Definition einer Anmeldezuordnung – sofort zur Verfügung.

Wurde eine Verbindung unterbrochen oder wird sie nicht automatisch beim Anmelden neu aufgebaut, kann die bereits erstellte Verbindungsdefinition verwendet werden, um über die Schaltfläche *Wiederherstellen* die in der Liste ausgewählte Verbindung erneut aufzubauen.

Um eine Verbindungsdefinition zu ändern, ist diese in der Liste der verfügbaren Verbindungen auszuwählen und über die Schaltfläche *Ändern* der Definitionsdialog erneut zu öffnen. Hierbei stehen wieder die Optionen, ob diese Verbindung nach dem Anmelden automatisch wieder aufgebaut werden soll beziehungsweise ob vor dem Verbindungsaufbau ein Paßwort angegeben werden soll, zur Verfügung. Auch können auf dieser Seite bestehende Verbindungen unterbrochen oder gar eine bestehende Verbindungsdefinition gelöscht werden.

Die weiteren Seiten des Notizbuchs *Gemeinsame Ressourcen und Netzwerkverbindungen* sind für die Konfiguration des lokalen Rechners als Peer-Server sowie die Ressourcen und Rechtevergabe verantwortlich und werden im Kapitel »OS/2 Warp V4 in einem Peer-Netzwerk« detailliert besprochen.

Besitzt man die entsprechenden Berechtigungen innerhalb der Domäne, kann man die Benutzerverwaltung vom OS/2-Arbeitsplatz ausführen. Einzelheiten hierzu finden Sie im Kapitel »Benutzerverwaltung«, in dem die einzelnen Möglichkeiten der Benutzerverwaltung besprochen werden.

Verwendung von Netzwerkdruckern

Ein Netzwerkdrucker stellt ein Druckerobjekt dar, das keine lokale Druckerwarteschlange besitzt. Bei einem Netzwerkdruckerobjekt wird ein Druckauftrag direkt in die Druckerwarteschlange auf dem Server gespoolt. Im Gegensatz dazu wird bei einem lokalen Druckerobjekt, dessen Anschluß (z.B. LPT4) einer Druckerressource im Netzwerk zugeordnet ist (umgeleiteter Anschluß), zuerst in die lokale Druckerwarteschlange gespoolt und anschließend erst in die Druckerwarteschlange des Druckers im Netzwerk.

Die Verwendung eines Netzwerkdruckers entbindet jedoch nicht von der Notwendigkeit, den richtigen Druckertreiber lokal zu installieren. Hier ist es von Vorteil, wenn der Netzwerkadministrator die entsprechenden Druckertreiber über das Netzwerk zur Verfügung stellt. Ansonsten blieb einem bisher nur, die Disketten oder die CD zu bemühen. Doch es geht auch noch anders. OS/2 Warp V4 bietet eine Druckertreibersynchronisation.

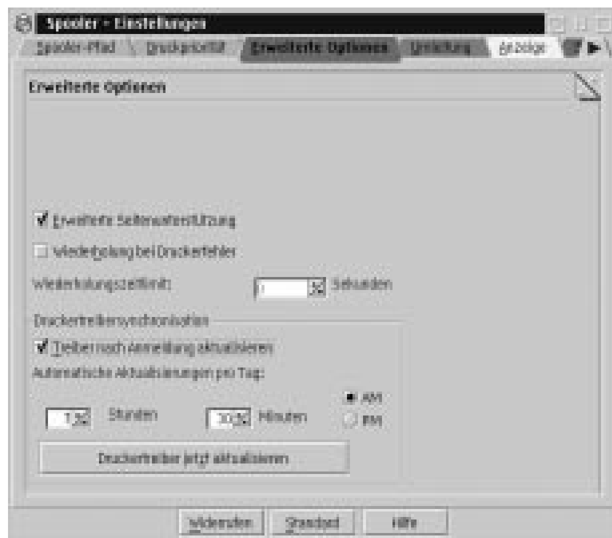


Abb. 9.109: Einstellungen für die Druckertreibersynchronisation

Hierbei holt sich der Client-Rechner

– nach der Anmeldung

– automatisch zu einem bestimmten Zeitpunkt

– auf Anforderung

die aktuelle Version des Druckertreibers vom Server ab. Diese Option versteckt sich in den Einstellungen für den Spooler, die sich wiederum im Ordner *Systemkonfiguration* finden. In den Einstellungen des Spoolers können auch die umgeleiteten Anschlüsse für Drucker verwaltet werden.

Um nun einen Netzwerkdrucker einzurichten, kann aus dem Ordner *Drucker*oder aus dem Ordner *System|Schablonen* eine Schablone verwendet werden. Im folgenden Dialog ist das zu verwendende Netzwerk, der Server sowie der Alias auszuwählen. Ist der richtige Druckertreiber bereits auf dem Rechner lokal installiert, ist die Konfiguration des Netzwerkdruckerobjekts damit abgeschlossen. Ansonsten fordert das System zur Installation des Druckertreibers auf.

Über das Kontextmenü des Netzwerkdruckerobjekts kann dieser einem lokalen Anschluß zugeordnet werden. Dies ist vor allen Dingen bei der Verwendung von DOS- und Windows-basierten Anwendungen erforderlich, da diese in der Regel nicht in der Lage sind, auf eine Druckerwarteschlange zu drucken. Ferner kann über das Kontextmenü eines Netzwerkdruckerobjekts auch auf einen anderen Netzwerkdrucker zugegriffen werden (*Zugriff auf anderen Netzwerkdrucker*).

Öffnet man das Notizbuch mit den Einstellungen, können noch weitere Einstellungen für das Netzwerkdruckerobjekt getroffen werden. Auf der Seite *Anzeige* kann eingestellt werden, ob

– alle Druckaufträge oder nur die eigenen angezeigt werden sollen

– nach wievielen Sekunden die Anzeige aktualisiert werden soll

Auch die angezeigten Objekteigenschaften lassen sich auswählen. Neben den standardmäßig ausgewählten Eigenschaften

– Icon

– Titel

– Status

– Job-ID

– Größe

– Datum

– Zeit und

– Eigentümer

können noch die Eigenschaften

– Objekt-Klasse

– Echter Name

– Anschluß

ausgewählt werden.

Von Interesse sind auch noch die Seiten *Warteschlange* sowie *Netzwerk-Status*. Auf der Seite *Warteschlange* kann die Priorität für den Ausdruck festgelegt werden. Eine höhere Priorität bewirkt eine erhöhte Druckperformance zu Lasten der Gesamtperformance des Systems. Hier muß jeder für sich entscheiden, welcher Performance er die höhere Priorität einräumt. Von

Interesse kann unter Umständen noch das Wahlfeld *Druckerspezifisches Format* sein. Wird dieses Wahlfeld aktiviert, gehen die Druckdaten als RAW-Druckdaten an den Drucker. Dies ist dann notwendig, wenn Druckaufträge von DOS- oder Windows-Applikationen nicht korrekt ausgedruckt werden. Dies ist jedoch eher die Ausnahme, so daß hier in der Regel keine Änderungen vorgenommen werden müssen.

Auf der Seite *Netzwerkstatus* können die Netzwerkeigenschaften des Netzwerkdruckers eingesehen werden. Hierzu gehören das verwendete Netzwerk, der Server, die Druckerwarteschlange, ob der Netzwerkdrucker verfügbar ist oder nicht sowie der eventuell zugeordnete lokale Druckerport bzw. »Keins«, wenn keine lokale Schnittstelle zugeordnet wurde.

Ist man als Administrator angemeldet, erscheint im Kontextmenü des Netzwerkdruckerobjekts ein weiterer Eintrag *Ferne Verwaltung*. Über diesen Menüeintrag ist es möglich, auf dem Server einen neuen Drucker einzurichten, einen bestehenden zu kopieren oder einen eingerichteten Drucker zu löschen.

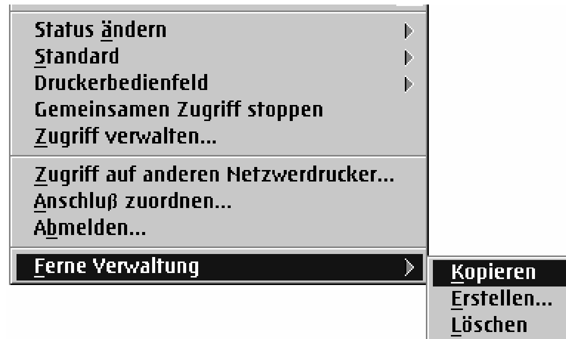


Abb. 9.110: Netzwerkdruckermenü für Administratoren

Wurde ein neuer Drucker auf dem Server eingerichtet, kann dieser anschließend ebenfalls »remote« für eine Aliasdefinition verwendet und im Netzwerk zur Verfügung gestellt werden. Dies kann entweder über die Befehlszeilen-Kommandos (NET ALIAS) oder über die grafischen Administrationstools (siehe unten) erfolgen.

Die grafischen LAN-Server-Administrationstools

Bei der Installation der Netzwerkkomponenten kann für die LAN-Requester-/Peer-Dienste angegeben werden, ob die grafischen LAN-Server-Admnistrationstools NETGUI installiert werden sollen. Wurde diese Option ausgewählt, findet sich im Ordner *Verbindungen|Netzwerkdienste* das Programm *LAN Server Verwaltung*. Dieses Programm wurde mit der Version 4 des IBM LAN Servers eingeführt und erlaubt eine grafische, objektorientierte Administration von LAN-Server-Domänen, der Benutzer- und Gruppenkonten von Windows-NT-Domänen sowie die Verwaltung von Peer-Arbeitsstationen.



Abb. 9.111: Die grafische LAN-Server-Verwaltung

Der obige Bildschirmausschnitt zeigt die LAN-Server-Verwaltung mit einer IBM-LAN-Server-Domäne (LS4DESKTOP), einer Windows-NT-Domäne (DOMAIN01) sowie einer Windows-95-Arbeitsgruppe (DESKTOP_SUPPORT). Die Arbeitsgruppe kann jedoch nicht administriert werden. Sie erscheint in der Anzeige, da in der »ibmlan.ini« der Arbeitsgruppenname beim Schlüsselwort *othdomain=* eingetragen wurde.

Da die Möglichkeiten der Administration sich für eine IBM-LAN-Server-Domäne, eine Windows-NT-Domäne und für den lokalen Rechner unterscheiden, wird im folgenden die Administration dieser drei Optionen im einzelnen behandelt.

Administration einer LAN-Server-Domäne

Zur Administration einer LAN-Server-Domäne stehen alle Möglichkeiten über die grafischen Administrationstools zur Verfügung. Öffnet man ein Domänenobjekt, finden sich weitere Objekte, die die Administration in logische Einheiten unterteilen. Hier stehen Ordner für:

– Gruppen

– Benutzereinträge

– Definierte Server

– Verzeichnisressourcen

– Druckerressourcen

– serielle Einheiten

– allgemeine Anwendungen

zur Verfügung.

Über die entsprechenden Ordner hat man dann Zugriff auf vorbereitete Schablonen, um ein entsprechendes Objekt innerhalb der Domäne zu erstellen. Um beispielsweise einen neuen Benutzer zu erstellen, ist der Ordner *Benutzereinträge* zu öffnen und aus der Schablone ein neuer Benutzer zu ziehen. Es öffnet sich sofort das Einstellungsnotizbuch für den neuen Benutzer. Hier können dann alle Einstellungen vorgenommen werden. Wurde dem neuen Benutzer z.B. keine Gruppen zugeordnet, so kann man den neu angelegten Benutzer mittels

Drag&Drop einer Gruppe zuordnen. Eine Besprechung der Administration des IBM LAN Servers würde den Rahmen dieses Buches sprengen, so daß an dieser Stelle auf die OnLine-Hilfe zur Administration einer LAN-Server-Domäne verwiesen werden muß.

Die Online-Hilfe zur Administration befindet sich im Ordner *Unterstützung|Information|Funktionen* und hier die Online-Dokumentation »LAN Server Administration Guide«.

Administration einer Windows-NT-Domäne

Öffnet man den Ordner einer Windows-NT-Domäne, stehen rein optisch die gleichen Objekte wie für eine LAN-Server-Domäne zur Verfügung. In einer Windows-NT-Domäne können jedoch nur die Benutzer- und Gruppenkonten verwaltet werden. Hierbei stehen jedoch ebenfalls nicht alle Optionen einer LAN-Server-Domäne zur Verfügung. Die Unterschiede treten jedoch erst zutage, wenn Aktionen durchgeführt werden, die von Windows NT nicht unterstützt werden. So ist es für eine Windows NT nicht möglich, Gruppen neue Benutzer mittels Drag&Drop zu übergeben. Wird dies versucht, so erhält man eine Fehlermeldung, die besagt, daß die ausgeführte Aktion von der API nicht unterstützt wird.

Auch wenn versucht wird, einen der Ordner für eine Ressourcendefinition zu öffnen, erhält man eine Fehlermeldung.

Für Benutzerkonten kann man:

– neue Benutzer anlegen

– Benutzer löschen

– Benutzer einer Gruppe zuordnen

– Passwort ändern/löschen

– Benutzerverzeichnisse festlegen

Für Gruppenkonten stehen folgende Optionen zur Verfügung:

– Gruppe anlegen

– Gruppe löschen

– Benutzer in eine Gruppe aufnehmen

– Beschreibung ändern

Bearbeitet man einen existierenden Benutzereintrag, muß man jedoch ständig die Meldung bezüglich nicht unterstützter API-Funktionen bestätigen. Dies macht die grafische Administration nicht gerade zum prädestinierten Werkzeug, um die Benutzer und Gruppen einer Windows-NT-Domäne zu verwalten. Hier eignet sich die Benutzerprofilverwaltung viel besser.

Eine Verwaltung der Freigaben einer Windows-NT-Domäne über die grafischen Administrationstools ist nicht möglich. Wird dies versucht, so erhält man ebenfalls eine Fehlermeldung, die besagt, daß die gewünschte Funktion nicht unterstützt wird.

Administration des lokalen Rechners

Für die Verwaltung des lokalen Rechners steht nur die Verwaltung der Netzwerkdienste zur Verfügung. Hierbei bezieht sich die Liste der verfügbaren Dienste auf die in der Sektion *Services* der »ibmlan.ini« gelisteten Dienste. Es wird pro Dienst Auskunft gegeben, ob der betreffende Dienst gestartet wurde oder nicht. Nach dem Starten des Rechners entspricht diese Anzeige den Einstellungen der »ibmlan.ini« unter dem Schlüsselwort *wrkservices=* und *srvservices=*. Über das Kontextmenü der Dienste kann ein gewählter Dienst angehalten bzw. gestartet werden. Wird ein Dienst geöffnet, erhält man nähere Informationen bezüglich der gestarteten Exe-Datei, dem Pfad und den beim Starten erhaltenen Rückkehrcodes.

Eine Gruppen- bzw. Benutzerverwaltung der lokalen Gruppen und Benutzer ist hier nicht möglich. Hierzu muß entweder auf die Benutzerprofilverwaltung, die Befehlszeile oder auf das Notizbuch *Gemeinsame Ressourcen und Netzwerkverbindungen* zurückgegriffen werden (siehe auch »Benutzerverwaltung«).

9.13.3 OS/2 Warp V4 als Windows-NT-Client

Eine Windows-NT-Domäne (Version 3.5x oder Version 4.0) präsentiert sich gegenüber einer OS/2-Warp-V4-Workstation wie eine IBM-LAN-Server-Domäne. Wird bei der Anmeldung für die zu verwendende Domäne eine gültige Windows-NT-Domäne angegeben und existiert in der Domäne das entsprechende Benutzerkonto, so kann von OS/2 Warp aus eine Anmeldung an der Windows-NT-Domäne erfolgen. Voraussetzung dafür ist jedoch, daß sowohl innerhalb der Domäne als auch auf dem OS/2-Warp-V4-Client das Protokoll NetBeui (NetBios) installiert und konfiguriert wurde.

Worin liegen nun die Unterschiede zwischen einer Integration in ein IBM-LAN-Server-Netzwerk und der Einbindung des OS/2-Arbeitsplatzes in eine Windows-NT-Domäne. Der offensichtlichste Unterschied für den Anwender, der sich sein Netzwerkverbindungen selbst erstellt, dürfte sein, daß in einer Windows-NT-Domäne keine Aliase zur Verfügung stehen. Freigaben in einer Windows-NT-Domäne (wie auch in einer Windows-NT-Arbeitsgruppe) müssen immer über den vollständigen UNC (Universal Naming Convention) in Form von »\\server\freigabe« angesprochen werden. Hier liegt der Verdacht nahe, daß die Anbindung von OS/2 Warp V4 an eine Windows-NT-Arbeitsgruppe vollkommen identisch zu einer Anbindung an eine Windows-NT-Domäne ist. Dem ist nicht so. Bei der Arbeit innerhalb einer Windows-NT-Arbeitsgruppe bewegt man sich in einem reinen Peer-Netzwerk. Dies hat zur Folge, daß auf jedem Rechner der Benutzer in der lokalen Benutzerdatenbank des Rechners existieren muß. Daraus ergeben sich die bereits bekannten Nachteile, daß die Verwaltung der Benutzer sehr viel aufwendiger ist als innerhalb einer Domäne. Die Benutzung bzw. Zuordnung einer freigegebenen Ressource einer Windows-NT-Domäne unterscheidet sich tatsächlich nicht von der Benutzung einer Ressource in einer Arbeitsgruppe, da in beiden Fällen der vollständige UNC-Name nötig ist. Lediglich die Validierung macht den kleinen, aber feinen Unterschied.

Doch zurück zur Integration des OS/2-Warp-Clients in die Windows-NT-Domäne aus Anwendersicht. Nach der erfolgreichen Anmeldung an die Windows-NT-Domäne kann sofort auf alle Freigaben innerhalb der Domäne zugegriffen werden. Im *LAN Requester Ressourcen Browser* werden alle Server der Windows-NT-Domäne aufgelistet. Es gibt jedoch kein Objekt, das Aus-

kunft über alle Freigaben innerhalb der Domäne gibt, so wie es das Objekt *Aliasnamen für Domäne xyz* für eine LAN-Server-Domäne ermöglicht. Es wird zwar ein ein entsprechendes Objekt angezeigt, doch wenn man versucht, es zu öffnen, erscheint die Fehlermeldung, daß die angeforderte API am fernen Server nicht unterstützt wird und es öffnet sich ein leerer Ordner. Stattdessen muß jedes einzelne Serverobjekt geöffnet werden, um die darin enthaltenen Freigaben zu sehen. Über das Kontextmenü einer Freigabe kann dann diese einer lokalen Einheit wie Druckerport oder Laufwerksbuchstabe zugeordnet werden beziehungsweise eine Abmeldung oder ein Zugriff auf einen anderen Server beziehunngsweise ein anderes Netzwerk (z.B. Novell NetWare) durchgeführt werden.



Abb. 9.112: Kontextmenü einer Windows-NT-Freigabe

Um nicht jedes Ordnerobjekt öffnen zu müssen, um die darin enthaltenen Freigaben sehen zu können, kann man auch an der Befehlszeile folgenden Befehl absetzen:

```
NET VIEW \\servername
```

Dadurch werden alle auf dem angegebenen Server *\\servername* vorhandenen Freigaben angezeigt. Dies ist vielleicht eine nicht mehr ganz zeitgemäße, aber sicher eine sehr schnelle Möglichkeit, alle Freigaben eines beliebigen Servers zu sehen. Wird »NET VIEW« ohne Angabe eines Servers aufgerufen, werden alle verfügbaren Server aufgelistet.

Um auch in einer Windows-NT-Domäne mit einer konsistenten Netzwerkumgebung zu arbeiten, empfiehlt es sich daher, die benötigten Netzwerkverbindungen als Verbindungsdefinitionen im Notizbuch *Gemeinsame Ressourcen und Netzwerkverbindungen* anzulegen (siehe Kapiel »OS/2 Warp V4 als IBM LAN Server Client«). Dadurch kann gewährleistet werden, daß beim Systemstart immer die identischen Verbindungen aufgebaut werden. Beim Erstellen einer Verbindungsdefinition ist jedoch immer der Radiobutton *Server* bei *Lokation der Ressource* anzugeben, da die Ressourcen ja über den vollständigen UNC-Namen eingetragen werden müssen. Zu beachten ist ferner, daß die Option *Serielle Einheit* für NT-Domänen ohne Bedeutung ist, da das Freigeben von seriellen Einheiten in Windows NT nicht unterstützt wird.

Alternativ zu dem Erstellen von Verbindungsdefinitionen kann auch serverseitig in der Windows-NT-Domäne der automatische Verbindungsaufbau geregelt werden. Hierzu muß im Benutzerprofil der Benutzerverwaltung ein Anmeldeskript angegeben werden, das die Netzwerkverbindungen aufbaut.

Ist man als Administrator (oder Konten-Operator) an einer Windows-NT-Domäne angemeldet, kann man die Benutzer und Gruppenkonten der Domäne vom OS/2-Arbeitsplatz aus administrieren. Hierzu können entweder die grafischen Administrationstools des IBM LAN Servers verwendet werden oder die *Dienste für Benutzerprofilverwaltung*, die sich im Ordner *System-*

konfiguration befinden, verwendet werden. Einige Funktionen der grafischen Administrationstools werden jedoch nicht unterstützt, da die entsprechenden Funktionen auf Windows-NT-Seite nicht zur Verfügung stehen (siehe oben). Besitzt man die entsprechenden Berechtigungen, auf einem im Netz vorhandenem anderen Windows-NT-Server zuzugreifen, kann man auch auf diesem Server die Benutzerverwaltung vom OS/2-Arbeitsplatz ausführen. Einzelheiten hierzu finden Sie im Kapitel »Benutzerverwaltung«, in dem die einzelnen Möglichkeiten der Benutzerverwaltung besprochen werden.

9.13.4 OS/2 Warp V4 als Novell-NetWare-Client

Wie bereits oben erwähnt, wird bei der Installation der Netzwerkkomponenten der Novell NetWare Requester V4.10 installiert, so wie er schon seit längerem im Einsatz ist. Einziger gravierender Unterschied ist, daß die Installationsroutine des NetWare Requesters für den Anwender nicht sichtbar ist. Da die Installation komplett über die Installationsroutine der OS/2-Warp-V4-Netzwerkkomponenten durchgeführt wird, wundert es auch nicht, daß die Treiberunterstützung der Netzwerkkarte auf dem NDIS-Treiber aufbaut, der für die anderen Netzwerkkomponenten benötigt wird. Der ODI-Treiber, der vom NetWare Requester installiert wurde, steht auch nicht in der »config.sys«, dafür wird der Schnittstellentreiber »odi2ndi.os2« installiert. Dieser regelt die softwaremäßige Emulation der ODI-Schnittstelle, wie sie von Novell verwendet wird, auf Basis eines NDIS-Treibers. Dadurch entfällt jedoch auch die Notwendigkeit, eine Konfigurationsdatei »net.cfg« für die Konfiguration der Netzwerkkarte anzulegen, da die Konfiguration der Netzwerkkarte über MPTS (Multi Protocol Transport Services) abgewickelt wird. Lediglich die Einstellung des *Preferred Servers* und ob die NDS (NetWare Directory Services) von Novell NetWare 4.x verwendet werden, werden in die Datei »net.cfg« eingetragen. Diese Datei wird im Root-Verzeichnis der Bootpartition von OS/2 angelegt. Die Datei »net.cfg« muß jedoch nicht zwingend existieren. Hier eine Beispieldatei für den Zugriff auf ein Bindery-basierendes Novell-Netzwerk, wobei der Server *AIBLING* der bevorzugte Server sein soll und keines der verwendeten NetWare-Protokolle (IPX/SPX) zusätzlich konfiguriert wurde.

```
NETWARE REQUESTER
   DIRECTORY SERVICES OFF
   PREFERRED SERVER AIBLING
```

Wird OS/2 Warp V4 als reiner Novell-NetWare-Client eingesetzt, kann man auch die »config.sys« dahingehend verändern, daß die Netzwerkanbindung nicht über den NDIS-Treiber, sondern direkt über den ODI-Treiber der Netzwerkkarte stattfindet. Soll nun die Netzwerkanbindung auf ODI umgestellt werden, sind alle Einträge, die das Verzeichnis »\ibmcom« im Pfad haben, zu entfernen. Ferner muß der »odi2ndi.os2«-Treiber gelöscht und stattdessen der richtige ODI-Kartentreiber eingetragen werden. Nach dem nächsten Systemstart wird die Novell-NetWare-Anbindung über den ODI Treiber abgewickelt.

```
REM -- NetWare Requester statements BEGIN --
SET NWLANGUAGE=ENGLISH
DEVICE=C:\NETWARE\LSL.SYS
RUN=C:\NETWARE\DDAEMON.EXE
DEVICE=C:\IBMCOM\PROTOCOL\ODI2NDI.OS2     => diese Zeile löschen
```

```
REM – ODI-Driver Files BEGIN –
rem DEVICE=C:\NETWARE\3C5x9.SYS          => Kommentarzeichen löschen
REM – ODI-Driver Files END –
DEVICE=C:\NETWARE\ROUTE.SYS
. . .
```

Achtung: Dadurch stehen alle anderen Protokolle sowie Netzwerkanbindungen nicht mehr zur Verfügung!!!

Wer mit dem direkten Editieren der Datei »config.sys« eher zurückhaltend ist, kann auch die Installationsroutine des NetWare Requesters aufrufen, um die Änderungen in der »config.sys« von der Installationsroutine des NetWare Requesters vornehmen zu lassen. Hierzu ist die Installationsroutine *NetWare-Client installieren* aus dem Ordner *Systemkonfiguration|Installieren/Entfernen* aufzurufen. Aus dem Menü *Installation* den Eintrag *Requester auf Arbeitsstation* auswählen. Nachdem das Ursprungs- und Ziellaufwerk bestätigt wurde, kann in dem darauf folgendem Dialog ausgewählt werden, welche Aktionen von der Installationsroutine durchgeführt werden sollen.



Abb. 9.113: Auswahl der Aktion der NetWare-Installation

Anschließend werden die Konfigurationsdialoge angezeigt, wobei der erste die Auswahl des zu verwendenden Netzwerkkartentreibers (ODI-Treiber) ermöglicht.

Soll eines der verwendeten Protokolle (z.B. IPX oder SPX) zusätzlich konfiguriert werden, kann man in der »net.cfg« die entsprechenden Abschnitte gemäß der Online-Dokumentation des Novell-NetWare-Requesters eintragen. Um dies etwas komfortabler zu gestalten und eine Hilfestellung durch exemplarische Beispiele zu erhalten, ist die Installationsroutine des NetWare Requesters zu starten und aus dem Menü *Konfiguration* und anschließend *Diese Arbeitsstation* zu wählen. Es öffnet sich ein Dialog, der den Pfad zu der Datei »net.cfg« abfragt und anschließend befindet man sich in einem Art Editor, der alle Optionen zur Konfiguration des NetWare Requesters bereithält. Auf der linken Seite des Programmfensters befinden sich alle gültigen Abschnittbezeichnungen sowie deren Schlüsselwörter. Im unteren Bereich kann man sich entweder eine Erklärung zu dem auf der linken Seite ausgewählten Eintrag oder ein Beispiel anzeigen lassen.

Doch wie gestaltet sich nun der Zugriff auf eine Novell NetWare Umgebung?

Prinzipiell gibt es keine nahtlose Integration einer Novell-NetWare-Umgebung in den LAN-Requester-Ressourcen-Browser für NETBIOS-basierende Netzwerke wie IBM LAN Server oder Peer-Verbindungen. Im Ordner *Verbindungen* erscheint stattdessen für NetWare-Netzwerke ein eigener Ordner *NetWare*. Ferner befindet sich im Ordner *Netzwerk Dienste* ein weiterer Ordner *NetWare Dienste*, der die Dienstprogramme für den NetWare Requester sowie die Online-Dokumentation enthält. Auch hier hat sich gegenüber dem Einzelprodukt des NetWare Requesters 4.10 nichts geändert.

Öffnet man im LAN-Requester-Ressourcen-Browser den Ordner *NetWare*, listet das System alle verfügbaren Server im Netzwerk auf.

Durch einen Doppelklick kann man nun den Zugriff auf einen Server starten. Hierzu ist eine Anmeldung an dem betreffenden NetWare-Server erforderlich. Liegt noch keine Anmeldung vor, erscheint der *Anmelden*-Dialog. Alternativ zum Doppelklick kann der *Anmelden*-Dialog auch über das Kontextmenü des Serverobjekts geöffnet werden. Anschließend werden alle Volumes und Druckerwarteschlangen des Servers angezeigt. Über das Kontextmenü dieser Objekte für Verzeichnisse beziehungsweise Druckerwarteschlangen kann nun das entsprechende Verzeichnis bzw. die gewählte Druckerwarteschlange einer lokalen Einheit zugeordnet werden. Ist das Verzeichnis beziehungsweise die Druckerwarteschlange bereits einer lokalen Einheit zugeordnet, kann die Zuordnung über das Kontextmenü aufgehoben werden. Auch ein Abmelden ist über das Kontextmenü der Netzwerkobjekte möglich. Öffnet man das Volume-Objekt eines Netware-Servers, kann man sich weiter durch den Verzeichnisbaum bewegen. Leider ist es in den Unterverzeichnissen jedoch nicht möglich, eine Laufwerkszuordnung (mapping) vorzunehmen, so daß man hier doch wieder auf die NetWare,Tools angewiesen ist.

Die NetWare-Tools

Im Ordner *NetWare Dienste* befindet sich das Programm *NetWare Hilfsprogramme*, über das Laufwerkszuordnungen sowie Druckerzuordnungen verwaltet werden können. Nach dem Programmstart wird eine Liste aller Laufwerke angezeigt. Hierbei werden in einem heterogenen Netzwerk alle NetBios-basierten Netzwerkverbindungen als lokale Laufwerke dargestellt. Das Laufwerk L: ist standardmäßig als Login-Laufwerk gemappt.

Bevor wir uns der Erstellung einer Netzwerkverbindung widmen, zuerst ein paar generelle Bedienungshinweise zu den NetWare-Tools. Wird in dem Menü *Werkzeuge* ein Eintrag ausgewählt, erscheint in der Menüleiste ein korrespondierendes Menü. In diesem finden sich dann alle Optionen zum gewählten Punkt. Zugleich wird für den gewählten Punkt ein eigenes Clientfenster geöffnet. Für die Menüeinträge *Plattenlaufwerke*, *Druckerschnittstellen* und *Servers* kann neben dem Menü auch noch über die Tasten [Einfg] bzw. [Entf] ein neuer Eintrag, sprich eine Server-Anmeldung, eine Drucker- bzw. eine Laufwerkszuordnung, erstellt bzw. gelöscht werden. Für alle verfügbaren Optionen kann über deren Menü der anzuzeigende Bereich festgelegt werden (z.B. nur alle gemappten Laufwerke vica alle Laufwerke).

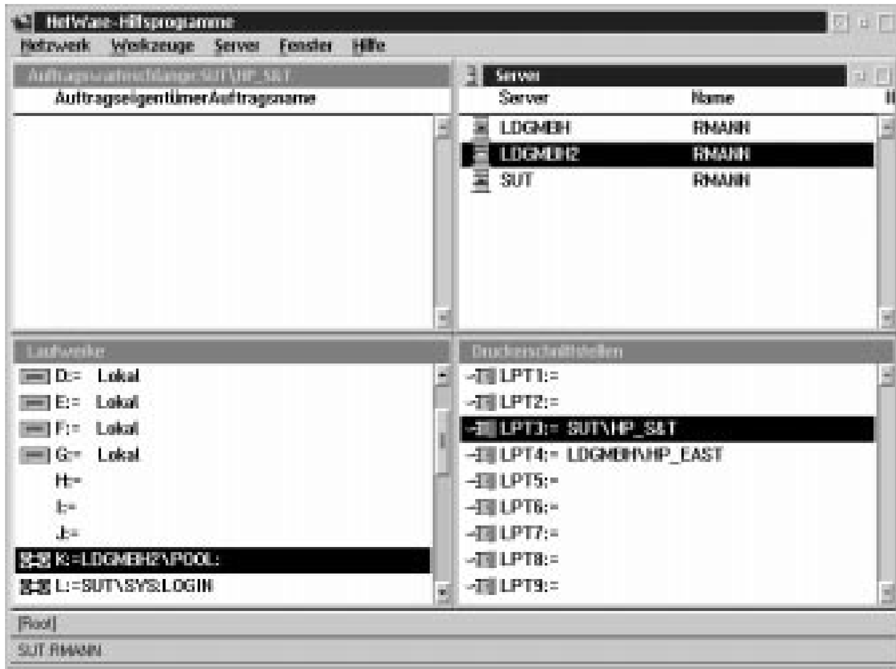


Abb. 9.114: Das Programm NetWare-Hilfsprogramme

Um eine Netzwerkverbindung zu erstellen, kann durch einen Doppelklick auf einen beliebigen Laufwerksbuchstaben der Dialog zur Definition einer Laufwerkszuordnung geöffnet werden. Hier kann dann der effektiv zu verwendende Laufwerksbuchstabe sowie die zu verwendende Netzwerkressource ausgewählt werden. In der Liste der zur Verfügung stehenden Netzwerkressourcen werden alle Server und deren Volumes gelistet, zu denen aktuell eine Verbindung besteht. Soll auf einen weiteren Server zugegriffen werden, ist über die Schaltfläche *Anschließen* die Dialogbox zur Anmeldung an einen weiteren Server zu öffnen, der Server auszuwählen und unter Angabe der Benutzerkennung und des Passwortes eine Anmeldung durchzuführen.

Anschließend erscheinen alle auf diesem Server vorhandenen Volumes in der Liste der verfügbaren Verzeichnisse. Der große Vorteil der NetWare-Tools bei der Laufwerkszuordnung liegt in der Möglichkeit, sich innerhalb der Verzeichnisstruktur eines Volumes zu bewegen und die Laufwerkszuordnung zum ausgewählten Verzeichnis vorzunehmen.

Ähnlich verhält es sich mit der Zuordnung einer Druckerwarteschlange zu einem lokalen Druckerport. Auch hier kann durch einen Doppelklick auf einen der verfügbaren Druckerports der Dialog zu Definition der Druckerzuordnung geöffnet werden. Hier stehen jedoch eine Reihe von Optionen zur Verfügung, wie die Druckerzuordnung konfiguriert werden soll. Abhängig von der Konfiguration der Netzwerkdruckerschlange kann eine gespeicherte Konfi-

guration abgerufen werden, die über das Programm »Printcon« vom Netzwerkadministrator definiert wurde. Ferner stehen folgende Optionen zur Verfügung:

– *Benachrichtigung wenn Auftrag gedruckt ist* sendet eine Nachricht an den Eigentümer des Druckauftrags, wenn dieser gedruckt wurde.

– *Formularvorschub* bewirkt, daß nach jedem Druckauftrag das Papier bis zum Beginn einer neuen Seite vorgeschoben wird. Da heutzutage nahezu alle Applikationen seitenorientiert drucken, kann diese Option deaktiviert werden.

– *Banner drucken* dient dazu, vor dem eigentlichen Druckauftrag eine zusätzliche Seite zu drucken, die Informationen bezüglich des Eigentümers des Druckauftrags, des verwendeten Druckerports sowie Datum und Uhrzeit enthält.

– *Kopien* legt fest, wieviele Kopien jeden Druckauftrags gedruckt werden sollen.

– *Dateiinhalt* wird verwendet, um die Art des Datenstroms zu bestimmen. Zur Wahl stehen »Bytestrom«, was für nahezu alle Anwendungsbereiche eingestellt werden sollte, sowie »Text«, was nur bei rein zeichenorientierten Druckausgaben eingesetzt werden sollte.

Tabulatorabstand wird verwendet, wenn bei *Dateiinhalt* der Wert »Text« angegeben wurde, um die Umsetzung eines Tabulators in die angegebene Anzahl Leerzeichen festzulegen

– Wurden für den Netzwerkdrucker Formulare definiert, so kann über das Drop-down-Feld *Format* das zu verwendende Formular ausgewählt werden. Ansonsten befinden sich keine Einträge in diesem Feld.

Um nun nicht jede Netzwerkverbindung zu einem Novell-NetWare-Netz bei jedem Systemstart neu aufbauen zu müssen, können die Netzwerkverbindungen zu Verzeichnissen bzw. Druckern in einer Datei gespeichert werden. Hierzu ist in den NetWare-Tools im Menü *Netzwerk* der Punkt *Einstellungen speichern* auszuwählen und die aktuellen Einstellungen in einer »*.nws« Datei zu speichern. Die so erstellte Datei mit den Verbindungen kann anschließend als Parameter für die NetWare-Hilfsprogramme eingegeben werden. Dadurch werden beim Starten der NetWare-Hilfsprogramme automatisch alle Verbindungen neu aufgebaut. Einziges Manko ist, daß die NetWare-Hilfsprogramme nicht im Hintergrund starten. Doch auch das kann mit einem kleinem Trick erreicht werden. Öffnet man das Notizbuch für das Objekt *NetWare-Hilfsprogramme* und löscht in der Programmzeile das letzte »e« von »nwtools.exe«, erkennt OS/2 den Programmtyp nicht mehr und man kann auf der Seite *Sitzung* das Wahlfeld *Beim Start Symbolgröße* aktivieren. Anschließend wechselt man wieder zur Programm-Seite und ergänzt in der Programmzeile das zuvor gelöschte »e« wieder. Dadurch wird erreicht, daß die NetWare-Tools beim Start vom System minimiert werden. Um das Ganze variabel zu gestalten, könnte man als Parameter für die »nwtools.exe« die eckigen Klammern »[ ]« angeben, damit das System beim Starten der NetWare-Tools nach dem zu verwendenden Parameter, sprich nach der zu verwendenden Konfigurationsdatei, fragt.

Damit ist es möglich, beim Starten des Systems immer eine konsistente Netzwerkumgebung vorzufinden. Es empfiehlt sich jedoch, in der »config.sys« den Parameter »Connections« in der Zeile *SET AUTOSTART=...* zu löschen. Ansonsten versucht OS/2, seinerseits alle beim Systemabschluß existenten Netzwerkverbindungen erneut aufzubauen.

Doch damit sind die Möglichkeiten der NetWare-Tools noch nicht erschöpft. Soll ferner das Programm »nwtools.exe« nach dem Aufbau der Verbindungen wieder geschlossen werden, kann als weiterer Parameter »autoexit« angegeben werden. Dadurch wird das Programm nach dem Einrichten der Netzwerkverbindungen wieder geschlossen. Jetzt stört eigentlich nur noch das NetWare-Logo, das beim Starten des Programms angezeigt wird. Aber auch hierfür gibt es einen Parameter, der die Anzeige des Logos unterdrückt. Hierzu ist lediglich das Wort »nologo« als weiterer Parameter zu übergeben. Eine vollständige Befehlszeile könnte dann wie folgt aussehen.

```
NWTOOLS.EXE mynet.nws nologo autoexit
```

Dadurch wird beim Starten der NetWare-Hilfsprogramme die Anzeige des Logos unterdrückt. Es erscheint der Anmeldedialog, in dem die Benutzerkennung und das Paßwort einzugeben sind. Anschließend werden alle in der Datei »mynet.nws« definierten Netzwerkverbindungen aufgebaut und das Programm beendet sich selbst.

Über das Menü »Tools« können noch weitere Aktionen innerhalb eines NetWare-Netzwerks durchgeführt werden.

Über den Menüpunkt *Server* ist es zum Beispiel möglich, sich alle Server, zu denen derzeit eine Verbindung besteht, oder auch alle im Netzwerk verfügbaren Server anzeigen zu lassen. Über die Tasten [Einfg] kann dann eine Anmeldung durchgeführt werden. Ferner kann hier auch das Paßwort für die Anmeldung an dem ausgewählten Server geändert werden. Wird zudem das Wahlfeld *Passwörter synchronisieren* aktiviert, durchsucht das Programm alle Server, zu denen momentan eine Anmeldung besteht, nach dem alten Paßwort. Wird dieses gefunden, erscheint eine Abfrage, ob das Paßwort für diesen Server durch das neue Passwort ersetzt werden soll. Somit ist auf einfachem Weg ein Passwortabgleich über mehrere NetWare-Server hinweg möglich.

Wird auf ein NetWare-4.x-Netzwerk zugegriffen, kann über den Menüpunkt *Werkzeuge|Verzeichnisbaum* der aktuelle Kontext bzw. der NDS-Server geändert werden. Diese Option ist nur dann sinnvoll wenn auch wirklich in einem NDS-Netzwerk gearbeitet wird.

Über *Werkzeuge|Benutzerlisten* können alle an einem Server angemeldeten Benutzer angezeigt werden. Wird diese Option gewählt, erscheint zuerst eine Dialogbox, in der der Server ausgewählt werden muß, dessen Benutzer gelistet werden sollen. In dieser Liste stehen nur die Server, zu denen aktuell eine Verbindung besteht. Über die Schaltfläche *Anschließen* kann jedoch jederzeit auf einen anderen Server zugegriffen werden. Dann kann an ausgewählte Benutzer eine Nachricht versandt werden.

Die letzte Option »Druckwarteschlangen« kann zur Verwaltung der Druckerwarteschlangen verwendet werden. Wird eine Druckerwarteschlange ausgewählt, öffnet sich wiederum ein Fenster, in dem alle Druckaufträge dieser Druckerwarteschlange gelistet werden. Abhängig von den Rechten ist es dann möglich, die eigenen Druckaufträge oder auch die Druckaufträge anderer zu verwalten.

Selbstverständlich ist es auch möglich, eine Netzwerkverbindung zu einem Novell-NetWare-Server als Freigabe anderen Benutzern im Netzwerk zur Verfügung zu stellen, selbst wenn diese nicht den NetWare Requester für Novell NetWare installiert haben. In diesem Fall agiert

der Rechner, der beide Requester – sowohl Novell NetWare als auch IBM LAN Server Requester – installiert hat, als Gateway. Die Rechner ohne NetWare Requester greifen auf eine solche Freigabe eines gemappten NetWare-Verzeichnisses wie auf jede andere Ressource innerhalb eines LAN-Server- oder Peer-Netzes zu. Es kann in diesem Fall auch der LAN-Requester-Ressourcen-Browser verwendet werden, um sich innerhalb der Freigabe zu bewegen. Hierbei benutzen alle die Benutzer-ID und das Paßwort des freigebenden Rechners (bzw. dessen Benutzers).

Neben den NetWare-Hilfsprogrammen befindet sich im Ordner noch das Programm *Netzwerkdrucker*, über das es möglich ist, Remote Printer für ein Novell-NetWare-Netz einzurichten sowie ein Novell-SMS- (Storage Management System) Agent, der es ermöglicht, die lokale Platte in eine Backup-Strategie zu integrieren.

Konfiguration des NetWare Requesters

Im Normalfall ist eine Konfiguration mit Ausnahme der bereits besprochenen Einträge nicht notwendig. Es gibt jedoch eine Reihe von Optionen zur Konfiguration des NetWare Requesters, die durchaus auch im täglichen Netzwerkbetrieb von Interesse sein könnten. Am einfachsten editiert man die Datei »net.cfg«, indem man die Installationsroutine des NetWare Requester aus dem Ordner *NetWare Dienste* startet und aus dem Menü *Konfigurieren* den Eintrag *Diese Arbeitsstation* auswählt.

Eines der ersten Probleme, in das man bei einer intensiv genutzten Netzwerkumgebung laufen kann, ist, daß der NetWare Requester standardmäßig das Loginlaufwerk auf den Laufwerksbuchstaben L: mappt. Soll das Loginlaufwerk einem anderen Laufwerksbuchstabe zugeordnet werden, sollte im Abschnitt *Netware Requester* in der »net.cfg« der Eintrag »Default Login Drive lw« (ohne Doppelpunkt) hinzugefügt werden. Anschließend sollten jedoch auch die Einträge in der »config.sys« unter LibPath, Dpath und Path angepaßt werden! Nach einem Neustart des Rechners wird das in der »net.cfg« angegebene Laufwerk als Loginlaufwerk verwendet.

Aus Performancegründen kann zudem die Verwendung von *Packet Burst* unterdrückt werden, wenn die eingesetzte Hardware (Netzwerkkarte) dies nicht unterstützt. Hierzu ist der Eintrag »Packet Burst off« unterhalb der NetWare-Requester-Sektion einzutragen. Des weiteren kann die Verwendung von *Large Internet Packages* ausgeschaltet werden, wenn keinerlei WAN-Verbindung vorhanden ist. Dies erledigt der Eintrag »Large Internet Packages off« ebenfalls unterhalb der NetWare-Requester-Sektion. Ferner kann es unter Umständen notwendig werden, den Protokoll-Stack für IPX und/oder SPX zu konfigurieren, wenn die eingesetzten Applikationen es erfordern. Dies würde hier jedoch zu sehr in die Protokolldetails gehen, so daß hier nicht weiter darauf eingegangen wird und auf die entsprechenden Programmdokumentationen und Netzwerkhandbücher verwiesen werden muß.

9.13.5 OS/2 Warp V4 in einem Peer-Netzwerk

Seit der Einführung von Windows für Workgroups erfreuen sich Peer-Netzwerke (oft zum Leidwesen der Administratoren) immer größerer Beliebtheit. Auch OS/2 Warp V4 spielt hier alle seine Fähigkeiten aus. Neben der Fähigkeit, Verzeichnisse, Laufwerke und Drucker frei-

zugeben, kann zwischen OS/2-Peer-Rechnern auch eine serielle Einheit (z.B. Modem) freigegeben werden. Diese kann jedoch nur von OS/2-Peer-Rechnern genutzt werden, da alle Windows-Varianten serielle Einheiten als Ressource nicht unterstützen. OS/2 Warp V4 unterstützt für die Freigabe von Ressourcen die Zugriffskontrolle auf Benutzerebene (User-Level Security). Dies bedeutet, daß eine Ressource prinzipiell für alle Benutzer freigegeben wird. Die Rechte, die der einzelne Benutzer für den Zugriff auf eine freigegebene Ressource besitzt, werden entweder direkt für den Benutzer, über seine Gruppenzugehörigkeit und die Rechte für die Gruppe oder pauschal für alle Benutzer festgelegt. Im Gegensatz dazu unterstützen Windows für Workgroups, Windows95 und die Peer-Dienste des IBM LAN Server Requesters auch die Ressourcenfreigabe auf Freigabeebene (Share-Level Security). In diesem Fall wird für die Freigabe ein Paßwort angegeben und jeder, der dieses Paßwort kennt, kann auf die Freigabe zugreifen. Ein weiterer gravierender Unterschied ist, daß bei der Freigabe einer Ressource mit Zugriffskontrolle auf Benutzerebene alle Benutzer bzw. Gruppen lokal definiert sein müssen, wohingegen bei einer Zugriffskontrolle auf Freigabeebene dies nicht erforderlich ist.

Standardmäßig werden die Peer-Dienste mitinstalliert und als automatisch zu startender Dienst in der »ibmlan.ini« konfiguriert (siehe Kapitel »Die IBMLAN.INI«). Startet man den Requester über »Net Start Req« oder über das Objekt *Starten LAN Requester-Dienste* im Ordner *Anmeldungen*, werden die Peer-Dienste mitgestartet und stehen sofort zur Verfügung, und einer Integration in ein bestehendes Peer-Netzwerk steht nichts mehr im Wege.

Dreh- und Angelpunkt für die Funktion als Peer-Rechner (Client und Server) ist das Notizbuch *Gemeinsame Ressourcen und Netzwerkverbindungen.* Über dieses Notizbuch können sowohl Freigaben anderer Rechner genutzt als auch eigene Ressourcen freigegeben werden. Die Definition einer Verbindung haben wir ja bereits im Kapitel »OS/2 Warp V4 als LAN Server Client« kennengelernt. Im folgenden sollen jedoch noch einmal alle Aspekte der Ressourcenfreigabe und Ressourcenverwendung innerhalb eines Peer-Netzwerks besprochen werden.

Freigeben von Ressourcen

Um anderen Rechnern den Zugriff auf eigene Ressourcen wie zum Beispiel Verzeichnisse, Drucker oder serielle Einheiten zu ermöglichen, muß die betreffende Ressource explizit für den Zugriff durch andere freigegeben werden. Um eine Ressource freigeben zu können, muß eine lokale oder Domäne-Anmeldung vorliegen, die den Benutzer als Administrator des lokalen Rechners identifiziert. Nur dann hat der lokale Benutzer die Autorität, um Ressourcen des lokalen Rechners frei zu geben. Eine freigegebene Ressource steht für die anderen Benutzer sofort im Netzwerk zur Verfügung.

Um eine Ressource freizugeben, gibt es drei Möglichkeiten. Die altmodischste ist, eine Ressource über die Befehlszeile unter Verwendung des Befehls »Net Share« freizugeben. Innerhalb einer RexX-Datei kann diese Methode jedoch durchaus noch Sinn machen. Um beispielsweise ein Verzeichnis »\public« für eine unbeschränkte Anzahl Benutzer unter dem Name »foryou« freizugeben und dem Benutzer »userID« das Lesen-Recht einzuräumen, sind folgende Befehle an der Befehlszeile abzusetzen:

```
NET SHARE FORYOU=C:\PUBLIC /REMARK:"Öffentliches Verzeichnis"
REM existieren noch Benutzerprofile für die Ressource, ist folgende Zeile zu
verwenden
```

```
REM NET ACCESS C:\PUBLIC /ADD userID:r
REM folgende Zeile ist nur gültig, wenn bereits ein Benutzerprofil angelegt
wurde
NET ACCESS C:\PUBLIC /GRANT userID:r
```

Die zweite und wesentlich komfortablere Methode, wenn es um das Einrichten einer einzelnen Freigabe geht ist, die Freigabe über das Laufwerks-, Verzeichnis- oder Druckerobjekt der freizugebenden Ressource einzurichten. Dazu ist nur das Kontextmenü des entsprechenden Objekts zu öffnen (rechte Maustaste) und aus dem Kontextmenü der Eintrag *Gemeinsamen Zugriff starten* auszuwählen. Es öffnet sich der folgende Dialog, über den der Freigabename sowie die lokale Ressource definiert werden kann. Zusätzlich wird noch die Option angeboten, diese Freigabe bei jedem Systemstart erneut einzurichten.

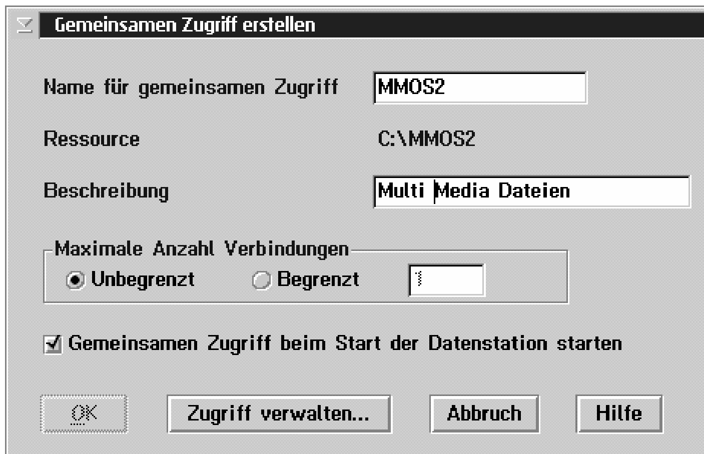


Abb. 9.115: Erstellen einer Freigabe

Ferner müssen über die Schaltfläche *Zugriff verwalten* sofort die Zugriffsrechte für die neu eingerichtete Ressource definiert werden. Hier bieten sich zwei Möglichkeiten an. Die Basis-Rechtevergabe und eine individuelle Rechtevergabe auf Benutzer und Gruppenebene. Die Basis-Rechtevergabe unterscheidet lediglich drei Arten von Zugriff:

– Kein Zugriff (die Vorgabe)

– Nur Lesen

– Lesen und Schreiben

Hier ist eine abgestufte Rechtevergabe nach Benutzern und/oder Gruppe nicht möglich. Dies ist auch nicht immer nötig. Soll beispielsweise ein CD-ROM Laufwerk freigegeben werden, ist eine pauschale Berechtigung »Nur Lesen« für alle Benutzer vollkommen ausreichend.

Soll eine nach Benutzern und/oder Gruppen abgestufte Rechtevergabe realisiert werden, ist der Radiobutton *Angepaßt* zu wählen. Hier erhält man dann einen nach Benutzern und Gruppen unterteilten Dialog, in dem dann für bestimmte Benutzer beziehungsweise Gruppen explizit Rechte vergeben werden können.

Aus Gründen der leichteren Verwaltbarkeit ist es jedoch zu empfehlen, die Rechte in erster Linie für Gruppen zu vergeben. Nur in Ausnahmefällen sollte ein Recht einem Benutzer direkt erteilt werden. Zu beachten ist ferner, daß die hier aufgelisteten Gruppen und Benutzer alle lokal definiert sind. Existiert eine identische Benutzerkennung innerhalb einer vorhandenen Domäne, sollte der Benutzer dafür sorgen, daß die von ihm verwendeten Passwörter in der Domäne als auch auf den Peer-Rechnern auf deren Ressourcen er zugreifen will, identisch sind.

Sind die Berechtigungen entsprechend den Erfordernissen zugewiesen worden, kann sowohl der Dialog zur Zugriffsverwaltung als auch der Dialog zur Definition der Freigabe mit »OK« verlassen werden und die anderen Benutzer in dem Netzwerk können sofort, gemäß ihren Berechtigungen, auf die neue Freigabe zugreifen.

Als letzte Möglichkeit steht noch das Notizbuch *Gemeinsame Ressourcen und Netzwerkverbindungen* zur Verfügung. Diese Möglichkeit bietet sich vor allem dann an, wenn mehrere Freigaben hintereinander definiert werden sollen oder auch dann, wenn auf eine Freigabe mit Zugriffskontrolle auf Freigabeebene eines Windows- oder LAN-Requester-Rechners zugegriffen werden soll. Im Notizbuch *Gemeinsame Ressourcen und Netzwerkverbindungen* können auf der Seite *Gemeinsame Ressourcen* nun die eigenen Ressourcen verwaltet werden. Es werden alle derzeit freigegebenen Ressourcen, sowohl Laufwerke, Drucker als auch serielle Einheiten mit ihrem aktuellem Status (ob die Freigabe gestartet wurde oder nicht) angezeigt. Die Spalte *Automatischer gemeinsamer Zugriff* gibt zudem Auskunft, ob die betreffende Freigabe sofort nach dem Starten der Peer-Dienste (des Requesters) freigegeben wird. Für eine in der Liste ausgewählte Freigabe kann die Freigabe beendet oder gestartet werden, abhängig von ihrem derzeitigem Status. Die Zugriffsdefinition kann über die Schaltfläche *Zugriff verwalten* aktualisiert werden. Um eine existente Freigabe zu ändern, zum Beispiel bezüglich der Option *Automatischer gemeinsamer Zugriff*, steht die Schaltfläche *Ändern* zur Verfügung, die den Dialog zum Erstellen einer Freigabe öffnet. In diesem Fall ist jedoch der Freigabename nicht editierbar.

Um eine neue Freigabe zu erstellen, ist über die Schaltfläche *Gemeinsamen Zugriff erstellen* der Dialog zum Anlegen einer neuen Freigabe zu öffnen. Hier kann nun zuerst gewählt werden, welche Art von Ressource freigegeben werden soll. Obwohl nur die Optionen *Verzeichnis*, *Drucker* und *serielle Einheit* in der Dialogbox angezeigt werden, kann selbstverständlich auch ein Laufwerk im Eingabefeld für *Verzeichnis* angegeben werden. Hierbei ist aber daran zu denken, daß ein jedes Laufwerk automatisch über die administrativen Freigaben für Administratoren freigegeben wird. Für Druckerfreigaben stehen alle im System bekannten Druckerwarteschlangen zur Verfügung, die durch ein Verzeichnis unterhalb des Verzeichnisses »\spool« repräsentiert werden. Diese Verzeichnisnamen entsprechen dem für das Druckerobjekt vergebenen physikalischem Namen (zu finden in den Eigenschaften eines Druckerobjekts). Für die Freigabe einer seriellen Einheit stehen automatisch alle 9 COM-Schnittstellen zur Verfügung, unabhängig davon, wieviele COM-Schnittstellen physikalisch im Rechner existieren.

Wurde die gewünschte Ressource für die zu erstellende Freigabe gewählt und die Dialogbox mit *OK* verlassen, gelangt man in einen weiteren Dialog, in dem man die Freigabeoptionen festlegen kann sowie auch den Zugriff auf diese neue Ressource.

Als erstes kann der Freigabenamen angegeben werden. Ein Freigabename muß folgende Bedingungen erfüllen:

– er kann bis zu 8 Zeichen lang sein

– er darf keine eingeschlossenen Leerzeichen enthalten

– es können beliebige Zeichen mit Ausnahme der Zeichen . « / \ Ø [ ] ; : ! < > + = , ? * verwendet werden

– das erste Zeichen darf keine Zahl sein

– das letzte Zeichen darf kein »$«-(Dollar-) Zeichen sein

Die letzte Einschränkung muß jedoch relativiert werden. Es ist sehr wohl möglich, eine Freigabe mit abschließendem Dollarzeichen anzulegen, nur eben nicht über die grafische Oberfläche. Hier ist wieder die Befehlszeile gefragt. Dies hat auch seinen besonderen Grund. Alle Freigaben mit einem abschließenden Dollarzeichen sind sogenannte »versteckte« oder auch »administrative« Freigaben und können nicht auf einem anderen Rechner oder in irgendeiner Art von Browser angezeigt werden. Aus diesem Grund können diese administrativen Freigaben auch nicht über die grafische Oberfläche angelegt werden. Auch über den Befehl *NET VIEW \\rechnername* kann ein anderer Rechner im Netzwerk diese administrativen Freigaben nicht sehen. Es ist also erforderlich, den genauen Freigabenamen zu wissen, um darauf zugreifen zu können. Die administrativen Freigaben auf dem lokalen Rechner können jedoch mit dem Befehl *NET SHARE* angezeigt werden. Nach dem Starten des Requester werden automatisch die folgenden administrativen Freigaben zur Verfügung gestellt:

– IPC$ für Interprozess-Kommunikation

– Admin$ das Verzeichnis »\ibmlan«

Anmerkung:

Für alle Windows-NT-Server und alle Server in einer IBM-LAN-Server-Domäne werden automatisch beim Start alle Festplattenlaufwerke als administrative Freigaben zur Verfügung gestellt. Hierbei wird dem Laufwerksbuchstaben das Dollarzeichen angehängt (C$,D$ usw.).

Neben dem Freigabenamen kann eine Beschreibung für die Freigabe angegeben werden. Der Übersichtlichkeit wegen sollte hier ein Text eingegeben werden, der die Freigabe eindeutig umschreibt, so daß die Benutzer diese Freigabe eindeutig klassifizieren können. Danach beginnt der interessantere Teil der Freigabedefinition. Eine Freigabe kann sowohl für eine beliebige Anzahl an gleichzeitigen Verbindungen oder für eine bestimmte Anzahl an Verbindungen eingerichtet werden. Dies kann beispielsweise verwendet werden, um entsprechend der Anzahl vorhandener Lizenzen den Zugriff auf ein bestimmtes Programm zu kontrollieren. Wird die Freigabe für eine unbeschränkte Anzahl an Verbindungen eingerichtet, ist die effektiv zur Verfügung stehende Anzahl an Verbindungen abhängig von der Konfiguration der Peer-Dienste, wie sie über die Seite *Konfiguration* des Notizbuchs *Gemeinsame Ressourcen und Netzwerkverbindungen* erfolgte.

Optional kann die Freigabe automatisch gestartet werden, sobald die Peer-Dienste gestartet werden. Andernfalls ist es notwendig, die Freigabe explizit über das Notizbuch zu starten.

Um die Freigabe erstellen zu können, ist es zudem zwingend erforderlich, über die Schaltfläche *Zugriff verwalten* den Dialog zur Vergabe von Zugriffsrechten zu öffnen und einmal durch Anklicken einer Berechtigung diese zu bestätigen. Andernfalls steht der OK-Button nicht zur Verfügung. Optional kann man ausgehend von diesem Dialog zur Vergabe von Berechtigungen neue Benutzer und neue Gruppen anlegen. Dies ist eine der vier Möglichkeiten, Benutzer und Gruppen zu verwalten. Einzelheiten zur Verwaltung von Gruppen und Benutzern finden sich im Kapitel »Benutzerverwaltung«. Im Dialog zur Definition der Zugriffsrechte steht der OK-Button ebenfalls nicht sofort zur Verfügung. Es muß zumindest das vorbelegte Zugriffsrecht durch erneutes Anklicken bestätigt werden, bevor die Dialogbox mittels OK geschlossen werden kann. Anschließend kann auch die Definition der Freigabe mit OK beendet werden.

Zugriffsrechte für freigegebene Ressourcen

An dieser Stelle möchte ich darauf hinweisen, daß ein Rechteprofil für eine Ressource und nicht für eine Freigabe (Freigabenamen) erstellt wird. Als Ressource ist hier ein Verzeichnis, ein Laufwerk oder eine Druckerwarteschlange zu verstehen. Das bedeutet, daß nach dem Löschen einer Freigabe die Zugriffsberechtigung auf die Ressource nach wie vor existieren und unter Umständen bei einer erneuten Freigabe der gleichen Ressource unter einem anderen Freigabenamen zu berücksichtigen ist. Zudem kann auch über den Befehl »Net Access« eine Zugriffsberechtigung für eine Ressource definiert worden sein, bevor die Ressource über eine Freigabe zur Verfügung gestellt wurde.

Bei den zur Verfügung stehenden Zugriffsrechten für Freigaben kann zwischen zwei Optionen gewählt werden. Die erste Option wird als *Basis* bezeichnet und unterscheidet nur zwischen »Kein Zugriff«, »Lesezugriff« und »Lese-/Schreibzugriff«.

Diese Rechtevergabe ist für alle Benutzer identisch. Es ist keinerlei Abstufung möglich. Wird eine dieser Berechtigungen ausgewählt und mit OK bestätigt, warnt OS/2, falls bereits abgestufte Berechtigungen für die Ressource erteilt wurden. Existieren bei einer Verzeichnis-Freigabe Unterverzeichnisse, wird abgefragt, ob die ausgewählte Berechtigung auf alle untergeordneten Verzeichnisse angewandt werden soll. Dies überschreibt alle zuvor für die Verzeichnisse vergebenen Berechtigungen.

Soll eine differenziertere Zugriffkontrolle auf die freigegebene Ressource erfolgen, ist die zweite Option *Angepaßt* zu wählen. Dadurch werden in der Dialogbox zwei Listboxen sichtbar, in denen die lokal definierten Benutzer und Gruppen angezeigt werden. Hier kann man nun für ausgewählte Benutzer oder Gruppen die gewünschten Rechte vergeben.

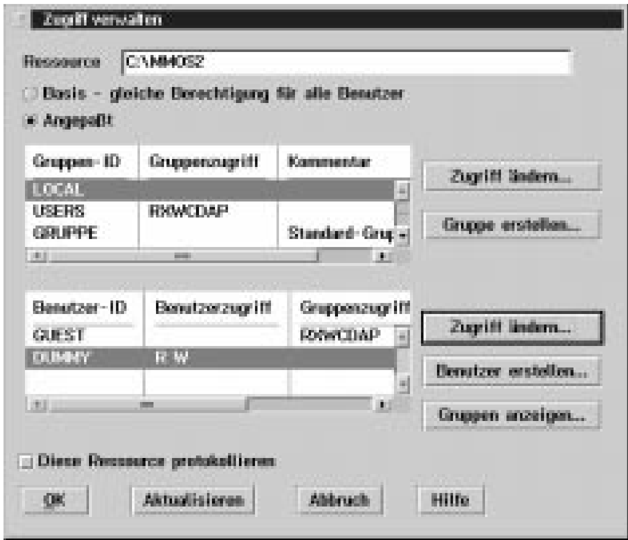


Abb. 9.116: Rechtevergabe für Benutzer und Gruppen

Hierzu wird über die Schaltfläche *Zugriff ändern* eine Dialogbox geöffnet, über die die zu vergebenden Rechte durch Anklicken der entsprechenden Wahlfelder ausgewählt werden können. Folgende Berechtigungen stehen zur Verfügung:

	Englische Bezeichnungen	**Deutsche Bezeichnungen**
R	Read	Lesen
W	Write	Schreiben
X	eXecute	Ausführen
C	Create	Erstellen
D	Delete	Löschen
P	Permission	Berechtigungen
A	Attribute	Attribute
N	None	Kein Zugriff

Hinweis: Befinden sich Batchdateien in einem freigegebenen Verzeichnis, ist das Ausführen-Recht nicht ausreichend, um diese Batchdateien auszuführen. Da eine Batchdatei vor der Ausführung zuerst komplett gelesen wird, müssen die Benutzer auch das Lesen-Recht auf dieses Verzeichnis besitzen.

Wurden die zu vergebenden Rechte ausgewählt und mit OK bestätigt, wird in diesem Fall nicht vor einem Überschreiben bereits existierender Berechtigungen gewarnt. In der Liste erscheinen dann die englischen Kürzel für die vergebenen Berechtigungen gemäß obiger Aufstellung. Es erscheint – bei existierenden Unterverzeichnissen – wiederum die Abfrage, ob die

erteilten Rechte auf die Unterverzeichnisse angewandt werden sollen. Werden die neuen Rechte nicht auf die Unterverzeichnisse angewandt, besteht an dieser Stelle die Möglichkeit, daß früher vergebene Rechte für den gewählten Benutzer oder die gewählte Gruppe die effektiven Rechte für die neue Freigabe beeinflussen. Hier muß genau überprüft werden, welche Berechtigungen für einen bestimmten Verzeichnisbaum vergeben wurden. Am einfachsten kann man dies an der Befehlszeile mittels des Befehls »Net Access *ressource* /Tree« in Erfahrung bringen. Dadurch werden für das in *ressource* angegebene Verzeichnis inklusive aller Unterverzeichnisse aller vergebenen Rechte für Benutzer und Gruppen angezeigt. Alternativ dazu kann man auf der Seite *Berechtigungen* des Notizbuches *Gemeinsame Ressourcen und Netzwerkverbindungen* sich alle Ressourcen auflisten lassen, für die eine ACL (*Access Control List*) erstellt wurde.

Es kann auch jederzeit für eine Ressource eine Berechtigung vergeben werden, ohne daß eine Freigabe erstellt wird. Hierzu kann das Kontextmenü der entsprechenden Ressource (Laufwerks- bzw. Verzeichnisobjekt oder Drucker) geöffnet werden und der *Zugriff verwalten*-Dialog geöffnet werden.

Ein weitere Möglichkeit, Rechte für Ressourcen zu verwalten, findet sich auf der Seite *Berechtigungen* des Notizbuchs *Gemeinsame Ressourcen und Netzwerkverbindungen*. Hier kann der gewünschte Ressourcentyp gewählt werden, um anschließend alle bereits vergebenen Rechte aufzulisten. Auch hier ist es wieder möglich, die Zugriffsrechte auf eine Ressource zu ändern, neu zu erstellen oder zu löschen.

Verwenden von Ressourcen

Der Zugriff auf eine Ressource innerhalb eines Peer-Netzwerks gestaltet sich denkbar einfach. Im LAN-Requester-Ressourcen-Browser werden alle Peer-Server, die in einer der unter *othdomain* gelisteten Domänen oder in der Standarddomäne gefunden werden, aufgelistet. Um die Freigaben eines Servers einzusehen, muß das Serverobjekt geöffnet werden. Alternativ können die Freigaben eines Servers auch wieder von der Befehlszeile mittels NET VIEW \\servername angezeigt werden. Anschließend kann über das Kontextmenü einer Freigabe diese einer lokalen Ressource (Laufwerksbuchstabe, Drucker- oder serieller Anschluß) zugeordnet werden oder die Freigabe weiter durchsucht werden, um die Ressource über den UNC-Namen zu nutzen. Hierbei ist jedoch zu beachten, daß im Serverobjekt nur die Freigaben auf Benutzerebene gelistet werden. Existiert eine Freigabe auf Freigabeebene (Share-Level Security), wie es von Windows für Workgroups, Windows 95 und den Peer-Diensten des IBM LAN Requester V4 unterstützt wird, so erscheint diese Freigabe nicht im Serverobjekt. Um auf eine solche Freigabe zugreifen zu können, muß entweder die Befehlszeile (z.B. »Net Use lw:\server\freigabenamen passwort«) oder das Notizbuch *Gemeinsame Ressourcen und Netzwerkverbindungen* benutzt werden.

Im Notizbuch *Gemeinsame Ressourcen und Netzwerkverbindungen* ist dazu auf der Seite *Verbindungen* der Dialog zum Erstellen einer Verbindung zu öffnen. In diesem Dialog ist dann das Wahlfeld *Kennwort anfordern* zu aktivieren, damit die Kennwortabfrage ausgeführt wird.



Abb. 9.117: Verbinden zu einer passwortgeschützten Freigabe

Wird eine passwortgeschützte Verbindung getrennt und erneut hergestellt, wird automatisch der Dialog zur Passworteingabe angezeigt. Besteht eine Verbindung zu einer passwortgeschützten Freigabe, erscheint diese Freigabe nach dem Verbindungsaufbau auch im Ordner des Serverobjekts und kann dort weiter durchsucht und bearbeitet werden.

Auch der Zugriff auf eine Windows-NT-Freigabe erfolgt über das Serverobjekt oder durch den *Verbindung erstellen*-Dialog des Notizbuchs *Gemeinsame Ressourcen und Netzwerkverbindungen*. Da Windows NT auch nur Freigaben auf Benutzerebene unterstützt, stellt sich das Problem mit dem Passwort hier nicht.

Konfiguration der Peer-Dienste

Um die Peer-Dienste von OS/2 Warp V4 zu konfigurieren, kann die Seite *Konfiguration* im Notizbuch *Gemeinsame Ressourcen und Netzwerkverbindungen* verwendet werden. Hier stehen zwar nicht alle Optionen für die Konfiguration der Peer-Dienste zur Verfügung, jedoch sind die Konfigurationsparameter, die am ehesten einer Änderung bedürfen, hier zu finden.

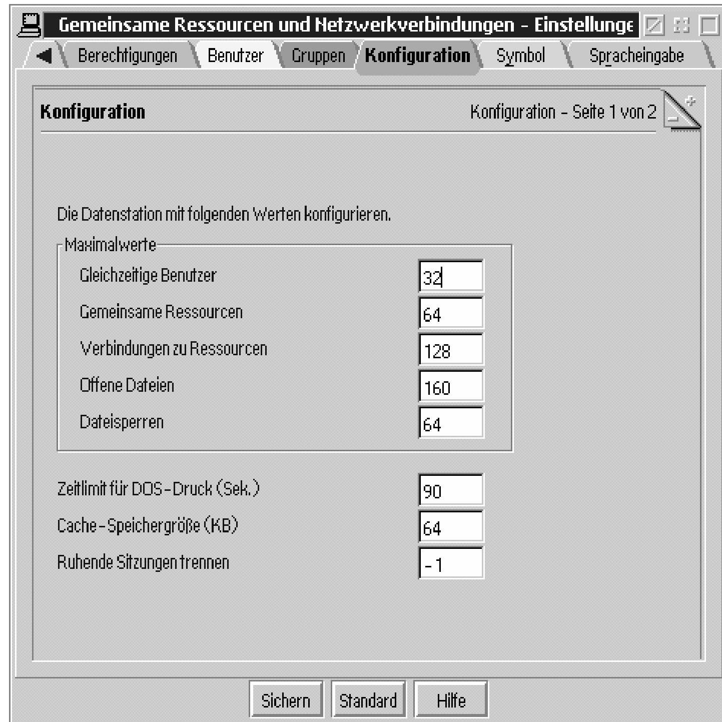


Abb. 9.118: Konfiguration der Peer-Dienste

Alle angezeigten Werte basieren auf den aktuellen Werten der »ibmlan.ini«. Wird einer der angezeigten Werte geändert, ist zum Sichern dieser Änderung zwingend die Schaltfläche *Sichern* zu benutzern. Wird das Notizbuch normal geschlossen, werden die Änderungen nicht in die »ibmlan.ini« zurück geschrieben.

Eine Beschreibung der zu ändernden Parameter findet sich im Kapitel »Die ibmlan.ini«.

Folgende Parameter können geändert werden.

- Gleichzeitige Benutzer: Korrespondiert mit *maxusers* in der »ibmlan.ini« und legt die Anzahl der gleichzeitigen Benutzer des lokalen Rechners fest.
- Gemeinsame Ressourcen: Korrespondiert mit *maxshares* in der »ibmlan.ini« und bestimmt die Anzahl der gleichzeitig benutzten gemeinsamen Ressourcen
- Verbindungen zu Ressourcen: Korrespondiert mit *maxconnections* in der »ibmlan.ini« und legt fest, wie viele gleichzeitige Verbindungen zu dem lokalen Rechner bestehen können.
- Offene Dateien: Korrespondiert mit *maxopens* in der »ibmlan.ini« und bestimmt die maximale Anzahl der gleichzeitig geöffneten Dateien

– Dateisperren: Korrespondiert mit *maxlocks* in der »ibmlan.ini« und legt die maximale Anzahl der gleichzeitig gesetzten Dateisperren fest

– Zeitlimit für DOS Druck: Korrespondiert mit *printbuftime* in der »ibmlan.ini« und legt fest, nach wievielen Sekunden ein Druckauftrag als beendet angesehen wird, wenn seitens der DOS-Applikation kein »End of job«-Zeichen gesendet wurde.

– Cache-Speichergröße: Korrespondiert mit *maxworkcache* in der »ibmlan.ini« und legt die maximale Größe des Übertragungspuffers fest.

– Ruhende Sitzungen trennen: Korrespondiert mit *autodisconnect* in der »ibmlan.ini« und gibt an, nach wieviel Minuten eine ruhende Verbindung zum lokalen Rechner getrennt werden soll.

Auf der zweiten Seite können die folgenden Parameter konfiguriert werden, die sich in erster Linie auf die Verwendung von seriellen Einheiten und Netzwerkparameter beziehen:

– Zeitlimit für COM-Öffnungen: Korrespondiert mit *charwait* in der »ibmlan.ini« und gibt an, wieviele Sekunden auf die Verfügbarkeit einer seriellen Einheit gewartet werden soll.

– COM-Sendeumfang: Korrespondiert mit *charcount* in der »ibmlan.ini« und legt fest, wie viele Bytes »gesammelt« werden sollen, bevor eine Übertragung der Daten an die serielle Einheit stattfindet.

– Zeitlimit für COM-Senden: Korrespondiert mit *chartime* in der »ibmlan.ini« und legt fest, wie lange (in Millisekunden) Daten zur Übertragung an eine serielle Einheit »gesammelt« werden sollen, bevor eine Übertragung stattfindet.

– Anzahl Netzwerkpuffer: Korrespondiert mit *numworkbuf* in der »ibmlan.ini« und bestimmt die maximale zur Verfügung stehende Anzahl an Übertragungspuffern.

– Größe: Korrespondiert mit *sizworkbuf* in der »ibmlan.ini« und bestimmt die Größe der Übertragungspuffer.

– Anzahl große Puffer: Korrespondiert mit *numbigbuf* in der »ibmlan.ini« und legt die maximal zur Verfügung stehende Anzahl an großen (64 Kilobyte) Transferpuffern fest.

9.13.6 Benutzerverwaltung

Um den Zugriff auf eigene Ressourcen besser steuern zu können, ist in OS/2 Warp V4 nur die Freigabe auf Benutzerebene möglich. Zu diesem Zweck ist eine vollständige Benutzerprofilverwaltung, wie man sie auch im IBM LAN Server findet, implementiert. Prinzipiell stehen vier Möglichkeiten der Benutzerverwaltung zur Verfügung.

Die einfachste Form ist die Verwaltung von Benutzern und Gruppen über die Befehle »Net User« und »Net Groups«. Da diese Art der Benutzerverwaltung keinerlei visuelle Unterstützung bietet und zugleich ein großes Maß an Lernaufwand bedeutet, um alle möglichen Parameter zu beherrschen, soll hier nicht näher darauf eingegangen werden. Der hauptsächliche Verwendungszweck der Benutzerverwaltung auf der Befehlszeile dürfte die Möglichkeit sein, die gewünschten Aktionen in ein ReXX-Skript zu integrieren, um einen gewissen Grad von Automatisierung zu erreichen.

Die zweite Möglichkeit, lokale Benutzer oder Gruppe zu verwalten, ist die Benutzerverwaltung im Notizbuch *Gemeinsame Ressourcen und Netzwerkverbindungen*. Hier findet man auf den Seiten *Benutzer* bzw. *Gruppen* eine Liste aller bereits in der lokalen Benutzerdatenbank (»net.acc«) vorhandenen Benutzer und Gruppen.

Auf der Seite *Benutzer* stehen die Optionen zum Aktualisieren, Erstellen oder Löschen eines Benutzers zur Verfügung. Für einen Benutzer gibt es an dieser Stelle leider keine Möglichkeit, seine Gruppenzugehörigkeit einzusehen. Hier muß auf die Befehlszeile oder auf die *Benutzereintragsverwaltung* zurückgegriffen werden.

Als Optionen für die Benutzerdefinition stehen folgende Punkte zur Verfügung:

- Benutzer-ID: Der Name, unter dem sich der Benutzer an dem Peer-Rechner anmelden kann. Dieser Name kann maximal 20 Zeichen lang sein und darf alle alphanumerische Zeichen mit Ausnahme der Zeichen « / \ [ ] ; : | < > + = , ? *

enthalten. Existiert in einer angeschlossenen Domäne derselbe Benutzer und soll diesem in der Domäne ein Basisverzeichnis (Homedirectory) zugewiesen werden, darf der Name nicht länger als 12 Zeichen sein!

- Beschreibung: Hier kann optional eine Beschreibung des Benutzers eingetragen werden. Diese Beschreibung kann bis zu 48 Zeichen lang sein.
- Benutzertyp: Über den Benutzertyp kann festgelegt werden, welche Berechtigung der Benutzer auf dem lokalen Rechner besitzen soll. Hierbei stehen drei Typen zur Verfügung. Der normale Benutzer kann nur auf die Ressourcen des lokalen Rechners zugreifen, sofern ihm die benötigten Rechte auf die Ressourcen erteilt wurden. Wurde für einen Benutzer das Wahlfeld *Bediener für Benutzereinträge* aktiviert, kann dieser Benutzer auch von einem anderen Rechner aus über das Netzwerk die Benutzer und Gruppen des lokalen Rechners verwalten. Als letztes sind noch die Administratoren zu nennen, die vollen Zugriff auf den lokalen Rechner haben.
- Anmeldung: Diese Option dient dazu, einen Benutzer zu deaktivieren bzw. zu aktivieren. Wird beispielsweise in einem Netzwerk mit Zeitarbeitern gearbeitet, die immer identische Rechte haben, aber eben kein ständiger Mitarbeiter sind, so kann über diese Option die Benutzerkennung deaktiviert werden. In diesem Fall bleiben alle Rechtedefinitionen und Gruppenzugehörigkeiten erhalten. Ein Anmelden mit dieser Benutzerkennung ist dann jedoch nicht mehr möglich. Ein solcher deaktivierter Benutzer kann dann bei Bedarf jederzeit wieder aktiviert werden, ohne daß die Rechtevergabe und Gruppenzugehörigkeit neu aufgebaut werden muß.
- Optionen: Für einen Benutzer kann festgelegt werden, ob er ein Paßwort benötigt oder nicht. Soll kein Paßwort verwendet werden, ist der Radiobutton *Kennwort wahlfrei* zu wählen. Andernfalls muß in den Eingabefeldern der Gruppe *Kennwort* das Kennwort eingegeben werden. Soll zudem der Benutzer gezwungen werden, nach dem ersten Anmelden mit dem vom Administrator vorgegebenen Paßwort ein neues Paßwort festzulegen, muß man das Wahlfeld *Kennwort abgelaufen* aktivieren. Aktualisiert man die Definition eines bereits existierenden Benutzers, findet man zusätzlich unter den Eingabefeldern für das Paßwort das Wahlfeld *Kennwort ändern*. Dadurch ist man als Administrator oder als *Bediener für Benutzereinträge* in der Lage, einem Benutzer ein neues Paßwort vorzugeben.

Wurden alle erforderlichen Angaben gemacht, wird der Benutzer in der lokalen Benutzerdatenbank angelegt.

Die Verwaltung von Gruppen gestaltet sich genauso einfach wie die Verwaltung der Benutzer. Auf der Seite *Gruppen* werden alle lokal definierten Gruppen angezeigt. Hier kann man sich nun alle Mitglieder einer selektierten Gruppe anzeigen lassen, wenn man auf die Schaltfläche zum Ändern der Gruppeninformation klickt. Für eine Gruppe können lediglich die Beschreibung und die Gruppenmitglieder festgelegt werden.

Leider ist das Zuweisen von Benutzern zu Gruppen über einen Benutzereintrag im Notizbuch nicht möglich. Ansonsten läßt die Benutzer und Gruppenverwaltung etwas an Komfort vermissen, wenn es zum Beispiel darum geht, einem Benutzer gleich Rechte auf eine existierende Freigaben zu geben.

Auf ähnlichem Niveau bewegt sich die Benutzereintragsverwaltung, die sich im Ordner *Systemkonfiguration|Dienste für Benutzerprofilverwaltung* befindet. Hier kann man ebenfalls alle Benutzer und Gruppen der lokalen Benutzerdatenbank verwalten. Zusätzlich ist es hier jedoch möglich, sofern man in einem Peer- , Windows-NT- oder IBM-LAN-Server-Netz über entsprechende Rechte zur Verwaltung von Benutzern verfügt, die Benutzer und Gruppen eines anderen Peer-Rechners oder einer IBM-LAN-Server- bzw. Windows-NT-Domäne zu administrieren.

Wird die Benutzereintragsverwaltung gestartet, wird nach dem zu verwendenden Ziel gefragt (sofern man an einer Domäne angemeldet ist).

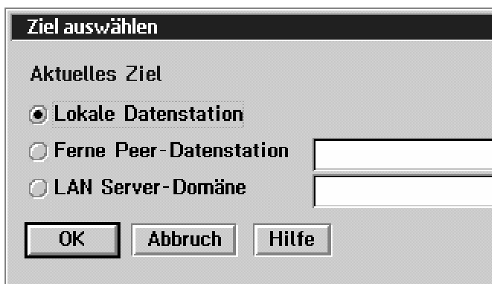


Abb. 9.119: Auswahl der zu bearbeitenden Benutzerdatenbank

Im eigentlichen Programm werden dann alle Informationen des aktuellen Benutzers angezeigt. Hierzu gehört die Information bezüglich des Benutzertyps, der Gültigkeitsdauer des Paßwortes, ob ein Paßwort benötigt wird und eine Liste aller Gruppen, in denen der aktuelle Benutzer Mitglied ist.

Über das Menü *Funktionen* können für den aktuellen Benutzer neben dem Kommentar auch noch das Paßwort sowie die Gruppenzugehörigkeit geändert werden.

Abhängig von der aktuellen Benutzerdatenbank stehen im Menü *Verwalten* die Optionen zum Verwalten der Benutzer bzw. Gruppen zur Verfügung. Wird eine dieser Optionen ausgewählt, erhält man sofort eine Liste aller entweder in der lokalen Benutzerdatenbank, auf dem fernen Peer-Rechner oder in der Domäne definierten Benutzer- bzw. Gruppeneinträge zur Bearbeitung.

Um einen neuen Benutzer zu erstellen (gleiches gilt auch für Gruppen), muß zwingend aus der Liste der vorhandenen Benutzer der Eintrag *Neu* ausgewählt werden. Ansonsten steht der Menüeintrag zum Erstellen eines neuen Benutzers oder einer neuen Gruppe nicht zur Verfügung. Diese Bedienung ist noch ein Relikt aus der LAN-Server-3.x-Zeit.

Bearbeitet man die lokale Benutzerdatenbank, kann zudem ein sogenanntes Benutzeranmelde-Profile verwaltet werden. Ein Benutzeranmelde-Profile ermöglicht es, bei lokaler Anmeldung automatisch an weiteren Rechnern angemeldet zu werden, wobei die zu verwendende Benutzerkennung und das zu verwendende Paßwort pro Rechner angegeben werden und demzufolge unterschiedlich sein kann. In einem reinen Peer-Netzwerk kann so – trotz unterschiedlicher Benutzeraccounts – auf den einzelnen Rechnern ein sogenanntes Single-Logon realisiert werden. Ist im Netzwerk eine LAN-Server-Domäne verfügbar, ist der Verwendung von Benutzeranmelde-Profilen die Verwendung der »Forwarded Authentication« vorzuziehen.

Um Zugriff auf die Benutzerdatenbank eines anderen Rechners oder einer Domäne zu erhalten, muß über das Menü *Funktionen|Ziel auswählen* das zu verwaltende Ziel gewählt werden. Ein »durchsuchen« nach verfügbaren Rechnern und Domänen ist nicht möglich. Wurde ein gültiger Domänen-Name oder Rechner-Name eingegeben, wird eine Verbindung zum Rechner aufgebaut und die Benutzerdatenbank des anderen Rechners oder der Domäne kann administriert werden.

Für die Administration der Benutzer und Gruppen für eine Windows-NT- oder IBM-LAN-Server-Domäne besteht noch die Möglichkeit, die Administration über die grafischen Administrationstools, wie sie mit IBM LAN Server 4.0 eingeführt wurden, durchzuführen. Hierzu ist die »LAN Server Administration« entweder über das entsprechende Icon *LAN Server Administration* im Ordner *Verbindungen|Netzwerkdienste* oder über den Befehl »netgui.exe« zu starten. Es öffnet sich der Ordner, in dem alle Domänen gelistet werden, auf die ein Zugriff besteht beziehungsweise die in der »ibmlan.ini« unter »othdomain=« gelistet wurden, plus der Standarddomäne. Auch eine Windows-NT-Domäne erscheint hier als Domäneneintrag sowie eine unter »othdomain=« eingetragene Windows-95- oder Windows-NT-Arbeitsgruppe. Eine Arbeitsgruppe kann jedoch nicht administriert werden.

Ferner findet sich hier noch ein Objekt, das den lokalen Rechner repräsentiert. Eine Benutzerverwaltung über dieses Objekt für den lokalen Rechner ist nicht möglich.

Für eine Windows-NT-Domäne können sowohl Benutzer als auch Gruppen verwaltet werden. Störend ist hierbei, daß bei jedem Wechsel der Seite im Notizbuch eines Benutzers oder einer Gruppe die Meldung erscheint, daß die angeforderte Funktion von der API nicht unterstützt wird. Dies macht einen produktiven Einsatz wenig sinnvoll. Somit bietet sich für die Administration einer Windows-NT-Domäne in erster Linie die Benutzereintragsverwaltung an. Ganz anders präsentiert sich die Benutzerverwaltung einer LAN-Server-Domäne über die grafischen Administrationstools. Hier kommt das objektorientierte Konzept von OS/2 voll zu tragen. Jeder Benutzer und jede Gruppe stellt ein Objekt dar. Daraus folgt, daß jederzeit ein Benutzerobjekt per Drag&Drop einer Gruppe zugeordnet werden kann und damit sofort ein Mitglied dieser Gruppe wird. Umgekehrt kann selbstverständlich auch eine Gruppe per Drag&Drop auf ein Benutzerobjekt gezogen werden und wird damit ebenfalls ein Mitglied dieser Gruppe. Für die Definition eines Benutzers stehen eine Reihe von Optionen zur Verfügung. Diese enthalten

neben den üblichen Vorgaben bezüglich Bemerkung und Paßwort auch Einstellungen für das Benutzer-Basisverzeichnis inklusive Festplatten-Beschränkungen, die Gruppenzugehörigkeit, die Anmeldezuordnungen, allgemeine Anwendungen sowie das zu verwendende Icon und den Objekttitel.

9.13.7 Netzwerk-Zwischenablage und Netzwerk-DDE

Im Rahmen der zur Verfügung stehenden Peer- und Requester-Dienste bietet OS/2 Warp V4 auch die Funktion der Netzwerk-Zwischenablage sowie die Möglichkeit, über Netzwerk-DDE eine rechnerübergreifende DDE-Verbindung aufzubauen. Hierzu muß auf den Rechnern, die den Inhalt der Zwischenablage austauschen wollen, das Programm »Netzwerk DDE und allgemeine Zwischenablage« (»nwdde.exe« im Verzeichnis »\ibmlan\netprog« gestartet sein. Nur dann kann eine Verbindung zwischen den Rechnern mit Zugriff auf die Zwischenablage aufgebaut werden. Wird das Programm gestartet, erscheint eine kleine PM-Anwendung mit einer Reihe von Schaltflächen, über die die zur Verfügung stehenden Aktionen ausgelöst werden können.

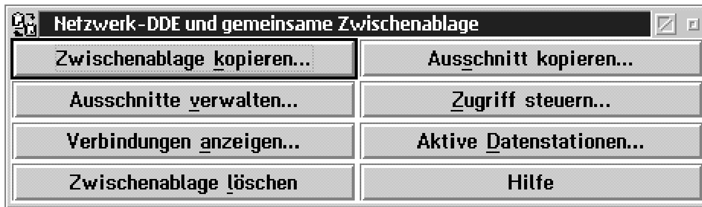


Abb. 9.120: Netzwerk-DDE und gemeinsame Zwischenablage

Zum Verständnis der Funktionsweise ist wichtig zu wissen, daß der Rechner der die Daten über die Zwischenablage zur Verfügung stellt, eher der passive Teilnehmer ist. Auf diesem Rechner sind die gewünschten Daten ganz normal von der Quellapplikation in die lokale Zwischenablage zu kopieren. Einzige Voraussetzung ist, daß die Anwendung *Netzwerk-DDE und gemeinsame Zwischenablage* gestartet wurde. Auf dem Zielrechner können anschließend über *Netzwerk-DDE und gemeinsame Zwischenablage* die Daten des Quellrechners aus dessen lokaler Zwischenablage in die lokale Zwischenablage des Zielrechners kopiert werden. Als letzter Schritt können dann die so übertragenen Daten in die Anwendung auf dem Zielrechner, unter Verwendung der normalen Bearbeitungsfunktionen *Einfügen* bzw. *Spezial einfügen* kopiert beziehumngsweise eine DDE-Verbindung aufgebaut werden. Auf diesem Weg ist der Datenaustausch zwischen OS/2-Rechnern und DOS-/Windows-Rechnern möglich. Die DOS-/Windows-Rechner müssen den IBM DOS LAN Requester installiert haben. Leider ist der Austausch der Zwischenablage mit Windows-Rechnern (WfW, Win95 und WIN NT) nicht möglich.

Innerhalb der Anwendung *Netzwerk-DDE und gemeinsame Zwischenablage* stehen folgende Funktionen zur Verfügung:

– *Zwischenablage kopieren* öffnet einen Dialog, über den der Quellrechner ausgewählt werden kann, dessen Zwischenablage über das Netzwerk in die lokale Zwischenablage kopiert wer-

den soll. Erscheint der gewünschte Rechner nicht in der Liste der verfügbaren Rechner, kann der NetBios-Name des Rechners im Eingabefeld eingegeben werden. Wird anschließend über die Schaltfläche *Kopieren* der Kopiervorgang gestartet, wird zuerst versucht, eine Verbindung zum angegebenen Rechner aufzubauen um – im Erfolgsfall – die Daten in die lokale Zwischenablage zu kopieren.

– *Ausschnitte verwalten* stellt den zentralen Punkt für die Bereitstellung von Ausschnitten dar. Ausschnitte sind Dateien, in denen der Inhalt der Zwischenablage – inklusive zusätzlicher Informationen wie DDE-Fähigkeit und Name des Ausschnitts – gespeichert werden. Befinden sich in der lokalen Zwischenablage Daten, können diese über die Schaltfläche *Ausschnitt sichern* als Ausschnitt gespeichert werden. Existierende Ausschnitte erscheinen in der Auswahlliste und können entweder gelöscht oder umbenannt werden. Über die Schaltfläche *Ausschnittdetails* kann abgefragt werden, ob die in diesem Ausschnitt gespeicherten Informationen DDE-fähig sind oder nicht. Die gleiche Information für die aktuellen Daten in der Zwischenablage erhält man über die Schaltfläche *Details zur lokalen Zw.ablage*. Die Verwendung von Ausschnitten bietet den Vorteil, daß unterschiedliche Daten zur gleichen Zeit allen Benutzern zur Verfügung gestellt werden können. Dem gegenüber kann bei der Verwendung der Daten aus der Zwischenablage immer nur ein- und dieselbe Information von mehreren Anwendern gleichzeitig benutzt werden. Zudem sind die Daten der Zwischenablage flüchtig, so daß ein Anwender nie sicher sein kann, ob der Inhalt der Zwischenablage, auf die er über das Netzwerk zugreift, nicht in der nächsten Sekunde geändert wird.

– *Verbindungen anzeigen* zeigt alle bestehenden DDE-Verbindungen zwischen anderen Rechnern und der lokalen Maschine an. Als Optionen stehen in diesem Dialog die Möglichkeit zum Trennen einer Verbindung sowie das Aktualisieren der Anzeige zur Verfügung.

– *Zwischenablage löschen* löscht den Inhalt der lokalen Zwischenablage.

– *Ausschnitt kopieren* öffnet einen Dialog, über den der Quellrechner ausgewählt werden kann. In diesem Fall wird jedoch auf dem Zielrechner nach vorhandenen Ausschnitten gesucht. Es findet jedoch keine echte Dateisuche statt. Vielmehr wird auf dem Quellrechner, der zuvor ausgewählt wurde, im Verzeichnis »\ibmlan\netprog\objbox« die Datei »resource.dir« ausgewertet. Diese Datei enthält die Namen der gespeicherten Ausschnitte sowie die Angabe des physikalischen Pfads zu den gespeicherten Ausschnittdateien. Diese werden ebenfalls im Verzeichnis »\ibmlan\netprog\objbox« unter dem Namen »cbox.bin« gespeichert. Hierbei wird für x eine fortlaufende Numerierung verwendet. Das Suchergebnis, respektive die Liste der in der Datei »resource.dir« gefundenen Namen der gespeicherten Ausschnitte, wird in der Listbox im unteren Bereich der Dialogbox angezeigt. Wird einer dieser Ausschnitte ausgewählt, kann man sich über die Schaltfläche *Ausschnittedetails* anzeigen lassen, ob die in diesem Ausschnitt gespeicherten Daten DDE-fähige Daten sind. Die Schaltfläche *In Zw.ablage kopieren* schließlich kopiert die auf dem Quellrechner in der ausgewählten Clippingdatei gespeicherten Daten in die lokale Zwischenablage, von wo aus sie weiter verwendet werden können.

– *Zugriff steuern* öffnet den Einstellungsdialog, über den festgelegt werden kann, in welcher Form andere Arbeitsstationen auf die Funktionen von *Netzwerk-DDE und gemeinsame Zwischenablage* zugreifen können. Hierbei können folgende Einstellungen getroffen werden:

– *Fernzugriff auf Zwischenablage möglich*: Ermöglicht/verhindert den Zugriff auf die lokale Zwischenablage des Rechners

– *Fernzugriff auf Ausschnitte möglich*: Ermöglicht oder verhindert den Zugriff auf die auf dem lokalen Rechner gespeicherten Ausschnittdateien. Effektiv wird nur das Auslesen der Datei »resource.dir« verhindert. Dadurch sind für andere Rechner keine Ausschnitte sichtbar.

– *Fernerstellung von DDE-Verbindungen möglich*: Ermöglicht oder verhindert den Aufbau einer DDE-Verbindung zwischen einem anderen Rechner und den lokal gespeicherten Daten. Diese Einstellungen sind generelle Einstellungen. Eine Aufschlüsselung der Zugriffsrechte nach Benutzern oder Gruppen ist nicht möglich.

– *Aktive Datenstationen*: Ermöglicht die Überprüfung eines NetBios-Namens auf dessen Fähigkeit, die Funktionen von *Netzwerk-DDE und gemeinsame Zwischenablage* zu nutzen. Erscheint die gewünschte Arbeitsstation nicht in der Liste der verfügbaren Rechner, kann dessen NetBios-Name im Eingabefeld eingegeben werden und über die Schaltfläche *Überprüfen* der Test gestartet werden. Konnte der Computer im Netzwerk gefunden werden, erscheint ein Statusfenster mit der Meldung, ob der angegebene Rechner die Funktionen von *Netzwerk-DDE und gemeinsamer Zwischenablage* unterstützt, und auf welchem Betriebssystem diese Funktionen auf dem angegebenen Rechner ausgeführt werden.

Netzwerk-DDE programmieren

Daß die DDE-Schnittstelle von *Netzwerk-DDE und gemeinsame Zwischenablage* voll unterstützt wird, manifestiert sich nicht zuletzt auch in der Möglichkeit, von einem beliebigen Programm aus über Makro- bzw. Script-Programmierung diese DDE-Schnittstelle zu nutzen.

Die innerhalb der anwendungsabhängigen Syntax der DDE-Befehle zu verwendende Syntax zur eindeutigen Bestimmung des DDE-Partners stellt quasi eine Verschachtelung der normalerweise verwendeten Syntax dar.

Normalerweise wird für eine DDE-Verbindung eine Applikation (Application) sowie ein Thema (Topic) verwendet. Innerhalb dieser Topics werden verschiedene Stellen (Items) angesprochen. Hierdurch ergibt sich folgender Aufbau für eine normale Befehlssequenz, um beispielsweise an ein bestimmtes Lesezeichen (Item) in einem in einer Textverarbeitung (Application) geöffnetem Dokument (Topic) einen Text per DDE einzufügen:

1. DDE-Verbindung initialisieren:

```
DDE-INITIATE Application, Topic
```

2. Per DDE Text an der Lesezeichen-Position einfügen:

```
DDE-POKE Application, Topic, Item, neuerText
DDE-TERMINATE
```

Um dies nun über das Netzwerk zu bewerkstelligen, ist die Vorgehensweise die gleiche, lediglich die Werte für *Application* und *Topic* sind statisch. Der Wert für *Item* stellt eine Zusammenfassung der normalen Reihenfolge Application – Topic – Item dar.

– Der Application-Name ist zwingend NWDDE.

– Der Topic-Name ist zwingend REMDDE.

Der Item-Name setzt sich zusammen aus:

– dem Namen des Rechners, auf der sich die Originaldaten befinden

– dem Namen des DDE-Servers auf diesem Rechner

– dem Topic-Namen der Datenquelle (in der Regel der Name der Datei oder »System«)

– dem Item-Bezeichner zur exakten Positionsbestimmung der Daten (z.B. ein Lesezeichen oder eine Zelladresse)

Hierbei ist die Syntax für das DDE-Item:

```
rechnername:application|topic#item
```

Somit ergibt sich beispielsweise folgender prinzipieller Aufbau für eine DDE-Verbindung zu einem Ami-Pro-Dokument »mydoc.sam« mit dem Lesezeichen »LZ1«, das auf dem Rechner »MyMachine« geöffnet wurde;

```
DDE-INITIATE NWDDE,REMDDE
DDE-POKE NWDDE,REMDDE,"myMachine":"AmiPro"|"MDOC.SAM"#"LZ1","neuer Text"
DDE-TERMINATE NWDDE,REMDDE
```

Die effektive implementierung hängt dann letztendlich von der verwendeten Script- bzw. Programmiersprache ab.

9.13.8 Der DHCP-Client- und DDNS-Manager

Wurde bei der Installation der TCP/IP-Komponenten angegeben, daß innerhalb des Netzwerks ein DHCP-Server zur Verfügung steht, kann die aktuell vom DHCP-Server zur Verfügung gestellte Adresse sowie die verbleibende Lease-Dauer über den *DHCP Client Monitor* eingesehen werden. Dieser *DHCP Client Monitor* findet sich im Ordner *Systemkonfiguration* .

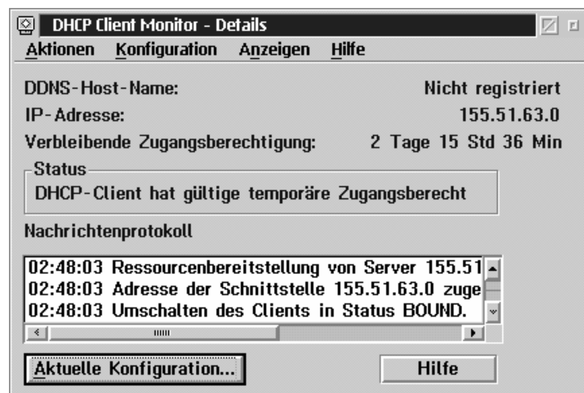


Abb. 9.121: Der DHCP-Client-Monitor

Der *DHCP Client Monitor* bietet zweierlei Ansichten. Zum einen die Übersichtsdarstellung, in der das Nachrichtenprotokoll ausgeblendet wird. In der Detail-Ansicht wird dieses Nachrichtenprotokoll angezeigt. Ferner kann noch die aktuelle Konfiguration eingesehen werden, in der neben den Informationen des Monitors noch die verwendete Netzwerkschnittstelle (z.B. »lan0«), die Teilnetzwerkmaske (SubNetMask) sowie der Domänenname angezeigt wird.

Wurde TCP/IP zusätzlich für die Verwendung von DDNS konfiguriert, kann man ebenfalls im Ordner *Systemkonfiguration* über das Programm *DDNS Konfiguration* die Konfigurationsparameter für DDNS angeben. Der dort spezifizierte Hostname wird dann an den DDNS-Server gesendet. Über diesen Namen ist dann der lokale Rechner im Netzwerk adressierbar.

9.14 MPTS – Multi Protocol Transport Services

von Raimund Mann

Ist es nach der Erstinstallation einmal nötig, die Konfiguration der Netzwerkkarte(n) und Protokolle zu ändern, stehen einem prinzipiell drei Möglichkeiten offen. Die erste wäre, mit einem normalen Editor die Datei »protocol.ini« zu laden und per Hand die Änderungen vorzunehmen. Dies sollte aber wirklich nur von sehr erfahrenen Anwender gemacht werden, die sehr fundierte Kenntnisse über die Protokolle und die »protocol.ini« besitzen. Ansonsten gilt hier wie für alle sensiblen Konfigurationsdateien erst einmal »Finger weg!«. Die zweite Möglichkeit besteht in dem erneuten Aufruf der Installationsroutine für die Netzwerkkomponenten »npinst.exe« (im Verzeicis »\ibminst« der Installations-CD). Hier kann man dann alle netzwerkrelevanten Komponenten erneut installieren und konfigurieren. Die letzte Möglichkeit besteht im Aufruf des Programms MPTS (»mpts.exe«), das sich im Verzeichnis »\ibmcom« befindet und nach der Installation von OS/2 Warp V4 im Ordner *Systemkonfiguration* abgelegt wird. Dieses Programm entspricht dem, das bereits seit LAN Server 4.0 verwendet wird und hat sich bestens bewährt.

Nach dem Start von MPTS wird zuerst gefragt, ob MPTS konfiguriert, vom Rechner entfernt oder ob eine neue Netzwerkkarte installiert werden soll. Letztere Option verbirgt sich hinter der Schaltfläche »Installieren«, sofern MPTS bereits auf der Maschine installiert wurde. Befindet sich noch kein MPTS auf dem Rechner, bewirkt die Schaltfläche *Installieren*, daß die MPTS-Dateien auf den Rechner kopiert werden.

Um nun Änderungen an der Konfiguration der Netzwerkkarte(n) und/oder Protokolle vorzunehmen, ist die Schaltfläche *Konfigurieren* zu wählen. Anschließend liest das Programm die aktuellen Konfigurationsinformationen aus der Datei »protocol.ini« aus.

Im folgenden Dialog ist nun die Option *LAN Adapter und Protokolle* zu wählen und erneut die Schaltfläche *Konfigurieren* auszuwählen. Dies öffnet das eigentliche Konfigurationsfenster, in dem alle unterstützten Netzwerkkarten sowie alle unterstützten Netzwerkprotokolle aufgelistet sind.

Zudem wird im unteren Bereich die aktuelle Konfiguration der verwendeten Netzwerkkarte(n) und Protokolle angezeigt. Um nun ein Protokoll zu löschen, ist dieses aus dem Fenster *Aktuelle Konfiguration* auszuwählen und über die Schaltfläche *Löschen* aus der Konfiguration zu entfernen. Umgekehrt kann ein Protokoll hinzugefügt werden, indem es im Auswahlfenster rechts oben in dem Dialog aus der Liste der verfügbaren Protokolle ausgewählt wird und über die Schaltfläche *Hinzufügen* die aktuellen Konfiguration ergänzt wird.

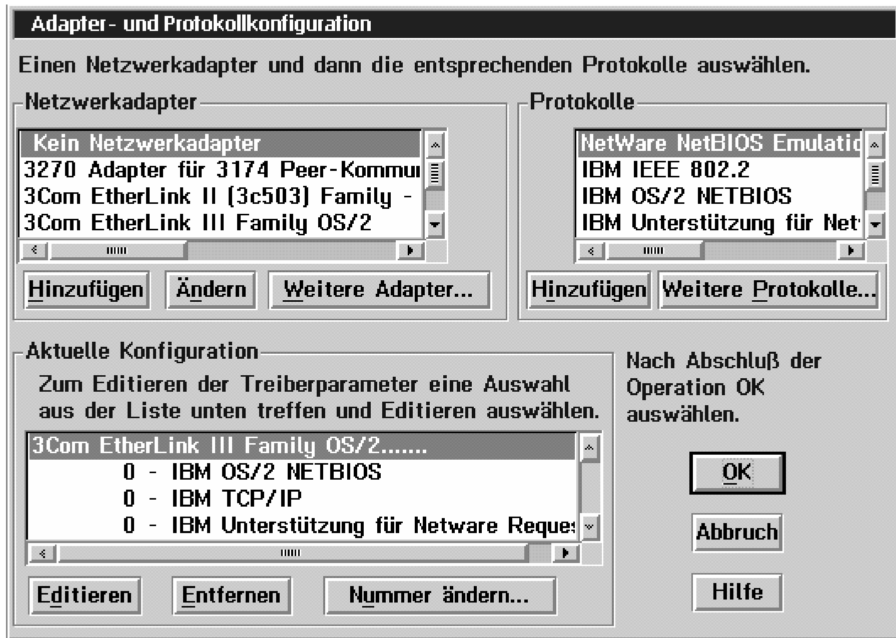


Abb. 9.122: Konfigurationsdialog von MPTS

Ist eine Konfiguration des Protokolls erforderlich (z.B. der Rahmentyp für die NetWare-Unterstützung), kann über die Schaltfläche *Bearbeiten* ein weiterer Dialog geöffnet werden, in dem alle Optionen direkt zur Bearbeitung zur Verfügung stehen. Die für ein Protokoll zur Verfügung stehenden Schlüsselwörter sind wie bei den Netzwerkkarten in »*.nif« Dateien hinterlegt. Diese befinden sich jedoch im Verzeichnis »\ibmcom\protocol« (z.B. die Datei »netbeui.nif« für die NetBios-Unterstützung). Im Normalfall ist hier jedoch kein Eingriff erforderlich. Daneben lassen sich für Protokolle logische Adapternummern vergeben.

Für die ausgewählte Netzwerkkarte lassen sich ebenfalls über die Schaltfläche *Bearbeiten* die Konfigurationsparameter setzen. Auch hier wird die zur Netzwerkkarte gehörende »*nif«-Datei (im Verzeichnis »\ibmcom\macs«) ausgelesen, um die darin definierten Schlüsselwörter im Dialog anzuzeigen.

Soll eine weitere Netzwerkkarte zum Einsatz kommen, ist aus der Liste der unterstützten Netzwerkkarten die entsprechende auszuwählen. Über *Hinzufügen* wird die Auswahl in die

aktuelle Konfiguration übernommen. Sind mehrere Netzwerkkarten installiert, kann festgelegt werden, welches Protokoll an welche Karte gebunden werden soll. Dazu ist lediglich die Netzwerkkarte oder ein bereits der Karte zugeordnetes Protokoll im Fenster *Aktuelle Konfiguration* auszuwählen und das gewünschte Protokoll hinzuzufügen.

Auf diese Weise ist es möglich, einen physikalisch getrennten Zugriff auf unterschiedliche Netzwerke zu realisieren. So kann beispielsweise die erste Karte mit dem Protokoll IBM NetBios für den Zugriff auf eine LAN-Server-Domäne konfiguriert werden und die zweite Karte für den Zugriff auf eine Novell-NetWare-Umgebung mit dem Protokoll IBM NetWare Requester Support. Selbstverständlich ist es auch möglich, ein Protokoll auf mehrere Karten zu binden. Dies wird für einen Client-Rechner sicherlich die Ausnahme sein.

Wird MPTS mit OK verlassen, werden die Konfigurationsparameter in die Datei »protocol.ini« geschrieben. Ferner wird, falls erforderlich, die »config.sys« aktualisiert. Hierzu muß beim Verlassen von MPTS das verwendete Bootlaufwerk angegeben werden. Die Aktualisierung kann je nach Rechner einige Zeit in Anspruch nehmen. MPTS meldet dann, ob die Aktualisierung der »config.sys« erfolgreich war oder nicht. Konnte die »config.sys« nicht aktualisiert werden, ist es schwer, die genaue Fehlerursache zu ermitteln. Eine Möglichkeit wäre, daß die Datei von einem anderen Programm (z.B. einem Editor) verwendet wird. Ist dies nicht der Fall, hilft es in der Regel, den Rechner neu zu starten und anschließend MPTS erneut aufzurufen. Es sollte dabei kontrolliert werden, ob die zuvor getroffenen Einstellungen in die »protocol.ini« geschrieben wurden. Das bedeutet, daß in den Konfigurationsdialogen der einzelnen Netzwerkkarten und Protokolle die neuen Werte stehen müssen. Anschließend kann MPTS verlassen werden. Nun sollte auch die Aktualisierung der »config.sys« erfolgreich durchgeführt werden können.

9.14.1 Die »protocol.ini«

Die Datei »protocol.ini« befindet sich im Verzeichnis »\ibmcom«. In dieser Datei werden alle Konfigurationsparameter zu Protokollen und Netzwerkkarten gespeichert. Beim Systemstart werden die darin enthaltenen Konfigurationsangaben verwendet, um die Netzwerkkarten zu initialisieren und die Protokolle an die entsprechenden Karten zu binden. Diese Datei ist eine reine ASCII-Datei, die von der Struktur her an eine Windows-3.x-INI-Datei erinnert. Es gibt Abschnitte, die durch Bezeichnungen in eckigen Klammern eingeleitet werden (vergleichbar den Sektionen in einer Windows-3.x-INI-Datei) und darunter befindet sich eine Reihe von Schlüsselwort-Wert-Kombinationen.

Hierbei kann man den Inhalt der »protocol.ini« prinzipiell in vier Bereiche unterteilen.

Der erste Bereich besteht nur aus einem einzigen Abschnitt und wird durch den Begriff »[PROT_MAN]« eingeleitet. Hier wird der zu verwendende Protokollmanager angegeben. Dies ist immer das in der »config.sys« eingetragene Programm »protman.exe«. Der Treibername wird über das Schlüsselwort *DRIVERNAME* angegeben. Dieser Abschnitt wird automatisch erstellt. Er beinhaltet keinerlei zu konfigurierenden Einträge.

```
[PROT_MAN]
    DRIVERNAME = PROTMAN$
```

Der zweite Bereich beinhaltet die MPTS-Konfigurationsinformation. Auch dieser Abschnitt wird automatisch erstellt. Hier findet eine Zuordnung von den Abschnittnamen für Protokolle und Netzwerkkarten zu den korrespondierenden »*.nif«-Dateien statt. Diese Zuordnung ist notwendig, damit MPTS die für die installierten Protokolle und Netzwerkkarten zur Verfügung stehenden Parameter, möglichen Werte und Hilfetexte ermitteln kann.

```
[IBMLXCFG]
    landd_nif = landd.nif
    netbeui_nif = netbeui.nif
    odi2ndi_nif = odi2ndi.nif
   PCMNICCS_nif = PCMNICCS.NIF
   elpcxos2_nif = elpcxos2.nif
```

Im nächsten Bereich finden sich die Netzwerkprotokolle wieder. Jedes Protokoll legt einen eigenen Abschnitt an, in dem die jeweiligen Parameter angegeben werden. Hierbei sind folgende Abschnitte möglich:

- [NETBIOS]: Konfiguriert das LM10-NetBios-Interface
- [landd_nif]: Konfiguriert das IBM-IEEE-802.2-Protokoll
- [netbeui_nif]: konfiguriert das NetBeui-Protokoll
- [odi2ndi_nif]: konfiguriert die NetWare-Gateway-Services
- [tcpip_nif]: konfiguriert das TCP/IP-Protokoll

Einzelheiten zu den jeweiligen Parametern der einzelnen Protokolle finden sich in der Online-Dokumentation wieder. Die entsprechende »*.inf« Datei heißt »mptscfg.inf« und befindet sich im Verzeichnis »\os2\book«.

```
[NETBIOS]
    DriverName = netbios$
    ADAPTER0 = netbeui$,0
 [landd_nif]
    DriverName = LANDD$
    Bindings = PCMNICCS_nif
    NETADDRESS = »I10005A9CF512"
    ETHERAND_TYP = »I"
    SYSTEM_KEY = 0x0
    OPEN_OPTIONS = 0x2000
    TRACE = 0x0
    LINKS = 8
    MAX_SAPS = 3
    MAX_G_SAPS = 0
    USERS = 3
    TI_TICK_G1 = 255
    T1_TICK_G1 = 15
   T2_TICK_G1 = 3
   TI_TICK_G2 = 255
   T1_TICK_G2 = 25
   T2_TICK_G2 = 10
   IPACKETS = 250
```

```
   UIPACKETS = 100
   MAXTRANSMITS = 6
   MINTRANSMITS = 2
   TCBS = 64
   GDTS = 30
   ELEMENTS = 800
   NETFLAGS = 0x0

[netbeui_nif]

   DriverName = netbeui$
   Bindings = PCMNICCS_nif
   NETADDRESS = »I10005A9CF512"
   ETHERAND_TYP = »I"
   USEADDRREV = »YES"
   OS2TRACEMASK = 0x0
   SESSIONS = 254
   NCBS = 255
   NAMES = 29
   SELECTORS = 15
   USEMAXDATAGRAM = »NO"
   ADAPTRATE = 1000
   WINDOWERRORS = 0
   MAXDATARCV = 4168
   TI = 30000
   T1 = 1000
   T2 = 200
   MAXIN = 1
   MAXOUT = 1
   NETBIOSTIMEOUT = 2000
   NETBIOSRETRIES = 3
   NAMECACHE = 1000
   RNDOPTION = 1
   PIGGYBACKACKS = 1
   DATAGRAMPACKETS = 10
   PACKETS = 335
   LOOPPACKETS = 8
   PIPELINE = 5
   MAXTRANSMITS = 6
   MINTRANSMITS = 2
   DLCRETRIES = 10
   FCPRIORITY = 5
   NETFLAGS = 0x0
```

Sind mehrere Netzwerkkarten (bis zu vier) im Rechner installiert, kann ein Protokoll auch an mehrere Netzwerkkarten gebunden werden. In diesem Fall werden alle Abschnittsnamen der Netzwerkkarten, auf die das Protokoll gebunden werden soll – durch Komma getrennt – beim Schlüsselwort Bindings angegeben. Ferner werden auch bei jedem Schlüsselwort des betreffenden Protokolls die gleiche Anzahl an Werten pro Schlüsselwort eingetragen. Im folgenden Auszug aus einer »protocol.ini« wird die Unterstützung für die NetWare-Gateway-Services ([odi2ndi_nif]) sowohl an die PCMCIA-Karte als auch an die 3COM Etherlink

```
III gebunden. Die PCMCIA-Karte wurde so konfiguriert, daß sie den Rahmentyp
802.3 unterstützt, wohingegen die 3COM-Karte den Rahmentyp 802.2 unterstützt.
[odi2ndi_nif]
   DriverName = odi2ndi$
   Bindings = PCMNICCS_nif,elpcxos2_nif
   NETADDRESS = »I10005A9CF512",
   TOKEN-RING = »no","no"
   TOKEN-RING_SNAP = »no","no"
   ETHERNET_802.3 = »yes","no"
   ETHERNET_802.2 = »NO","yes"
   ETHERNET_II = »NO","no"
   ETHERNET_SNAP = »NO","no"
   TRACE = 0x0
```

Die Reihenfolge der unter »Bindings« angegebenen Netzwerkkarten ist entscheiden für die logische Adapternummer. In obigem Beispiel wird die PCMCIA-Karte als Adapter 0 und die 3COM-Karte als Adapter 1 angesprochen. Soll aus irgendeinem Grund die 3COM-Karte als Adapter 2 angesprochen werden, ist unter dem Schlüsselwort »Bindings« ein Adapter unter Verwendung eines Kommas auszulassen. Die Zeile würde dann wie folgt lauten:

```
   Bindings = PCMNICCS_nif,,elpcxos2_nif
```

Im letzten Bereich wird die Konfiguration der installierten und konfigurierten Netzwerkkarten eingetragen. Hier findet sich mindestens der Treibername wieder. Ob weitere Parameter erforderlich sind, hängt von der verwendeten Netzwerkkarte ab. In dem folgenden Auszug aus einer »protocol.ini« wurden zwei Netzwerkarten konfiguriert, wobei die erste Karte eine PCMCIA-Karte (IBM Ethernet Credit Card Adapter) ist und die zweite eine 3Com Etherlink III.

```
[PCMNICCS_nif]
   DriverName = PCM_CS$
   IOBASE = 0x120
   INTERRUPT = 9
   RAMADDRESS = 0x0
   PCMCIA
[elpcxos2_nif]
   DriverName = ELPC3$
   MaxTransmits = 20
```

Werden mehrere Netzwerkkarten vom gleichen Typ verwendet, wird ebenfalls für jede Karte ein eigener Abschnitt angelegt. Ab der zweiten Karte wird jedoch eine fortlaufende Nummer, beginnend bei 2, angehängt. So ergeben sich beispielsweise folgende Aschnitte für drei identische Netzwerkkarten:

```
[elpcxos2_nif]
        ...
[elpcxos2_nif2]
        ...
[elpcxos2_nif3]
        ...
```

9.14.2 Die »ibmlan.ini«

So wie die »protocol.ini« für die Initialisierung der Netzwerkkartentreiber und Netzwerkprotokolle verwendet wird, wird die Datei »ibmlan.ini« (im Verzeichnis »\ibmlan«) verwendet, um den Requester zu initialisieren und zu konfigurieren. Wird der Requester entweder über das Icon *Starten LAN Requester Dienste* im Ordner *Anmeldungen* oder mittels »net start req[uester]« von der Befehlszeile aus gestartet, wird die »ibmlan.ini« ausgelesen und die darin enthaltenen Informationen verwendet, um die zu startenden Dienste sowie die Konfiguration der Dienste zu ermitteln. Das bedingt, daß nach Änderungen an der »ibmlan.ini« der Requester gestoppt und erneut gestartet werden muß. Wie bereits oben besprochen, lassen sich eine ganze Reihe von Parametern der Peer-Dienste (Peer-Sektion in der »ibmlan.ini«) über das Notizbuch *Gemeinsame Ressourcen und Netzwerkverbindungen* modifizieren. Es gibt jedoch durchaus auch Situationen, in denen ein manueller Eingriff in die Datei notwendig wird. Ein möglicher Grund wurde bereits im Kapitel »OS/2 Warp V4 als IBM LAN Server Client« besprochen (Hinzufügen weiterer Domänen). Soll die Standardvalidierung in der Domäne erfolgen, muß das 37. Bit der *wrkheuristics* in der Requester-Sektion auf den Wert 2 gesetzt werden.

Man sollte es sich jedoch folgendes zum Grundsatz machen, bevor Änderungen vorgenommen werden:

– Immer zuvor eine Sicherheitskopie anlegen

– Nur die Werte ändern, deren Funktion einem bekannt sind und die man versteht

– Keine Einträge aus der Datei löschen

– Nur einen Parameter ändern und dann die Auswirkung der Änderung überprüfen. Werden mehrere Parameter gleichzeitig modifiziert, ist es im nachhinein oft sehr schwer, festzustellen, welche Änderung die Verbesserung oder Verschlechterung bewirkt hat.

Die »ibmlan.ini« besteht ebenfalls aus Sektionen, in denen wiederum Schlüsselwort/Wert-Kombinationen gespeichert sind. Folgende Sektionen finden sich in einer »ibmlan.ini«:

– Networks

– Requester

– Messenger

– Peer

– Replicator

– Services

Obwohl es möglich ist, über die Befehlszeile mittels der Befehle »Net Start« und »Net Config« für eine Sitzung einige Werte zu überschreiben, dürfte dies wohl eher die Ausnahme sein.

Im folgenden werden die einzelnen Sektionen und deren wichtigsten Parameter besprochen.

Die Networks-Sektion

In der Networks-Sektion werden die installierten Netzwerkkarten, die zur Netzwerkanbindung zur Verfügung stehen, eingetragen. Für jede Netzwerkkarte ist ein eigener Eintrag notwendig, der mit »netx=« beginnt, wobei »x« eine fortlaufende Nummer von 1 bis 4 (jeweils einschließlich) ist.

```
netx = Treibername$, a, Treibertyp$, x1, x2, x3
```

Hierbei steht *Treibername$* für den zu verwendenden Protokolltreiber (NETBEUI$, TCPBEUI$, SNABEUI$ oder LOOPDRV$).

Der Wert a gibt die zu verwendende Netzwerkkarte an (0 bis 3). Dieser Wert muß mit dem Wert korrespondieren, der in der »protocol.ini« bei den »Bindings=«-Einträgen verwendet wurde. Hierbei legt die Position des zu bindenden Kartentreibers innerhalb der durch Komma getrennten Liste die logische Adapternummer fest (siehe auch »protocol.ini«).

Anschließend wird der Treibertyp angegeben (für TCPIP muß der Wert auf LM10 stehen).

Die Parameter x1, x2 und x3 sind im Gegensatz zu den obigen Systemparametern Parameter, über die die verwendbaren Ressourcen konfiguriert werden können.

- *x1*: Dieser Parameter legt die maximal zur Verfügung stehenden NetBios-Sessions fest, die auf dem Rechner verwendet werden können. Hierbei ist der maximale Wert 254. Jedoch darf der hier angegebene Wert nicht den Wert für *Sessions* in der Sektion *[netbeui_nif]* der »protocol.ini« überschreiten. Eine Session bezeichnet die Verbindung zwischen zwei Rechnern und nicht die Verbindung zu einer Ressource auf einem Rechner. So baut ein Rechner zu einem anderen eine Session auf, kann jedoch zugleich mehrere Verbindungen (Connections) zu Ressourcen auf diesem Rechner haben (z.B. durch mehrere Laufwerkszuordnungen). Der Wert muß mindestens 2 sein, da ansonsten kein Verbindungsaufbau möglich ist.
- *#x2*: Dieser Wert gibt an, wieviele NetBios-Befehle (Network Control Blocks, NCBs) gleichzeitig gesendet werden können. Auch hier gilt wieder, daß der angegebene Wert nicht größer sein darf als sein Pendant *ncbs* in der »protocol.ini«. Der Minimalwert ist 16.
- *x3*: Der letzte Parameter schließlich legt fest, wieviele NetBios-Namen gleichzeitig belegt werden können. Wie bei den beiden vorhergehenden Parametern darf auch hier der Wert nicht größer sein als der korrespondierende Wert unter *Names* in der Sektion *[netbeui_nif]* der »protocol.ini«. Der Wert muß mindestens 5 betragen.

Hier das Beispiel einer Konfiguration mit einer Netzwerkkarte, die das Protokoll NETBEUI auf der ersten Karte verwendet. Es wurden 100 Sessions, 150 NCBs und 14 (default) NetBios-Namen konfiguriert.

```
[networks]
 net1 = NETBEUI$,0,LM10,100,150,14
```

Wird der LoopBack-Treiber verwendet, um auf einem Rechner sowohl Server als auch Client zu simulieren, ohne Anschluß an ein physikalisches Netz zu haben, lautet der Eintrag;

```
net1=loopdrv$, 0
```

Die Requester-Sektion

Die Requester-Sekton dient der Konfiguration des IBM-LAN-Server-Requesters. Hierbei gibt es eine Reihe von Parametern, die in der Praxis nur in den seltensten Fällen eine Konfiguration benötigen.

Die wohl wichtigsten Parameter sind die Systemparameter. Dies sind die Parameter *computername*, *domain*, *othdomains*, *wrknets* und *wrkservices*.

- *computername*: Dieser Eintrag legt den NetBios-Namen des Rechners fest. Dieser Netbios-Name muß innerhalb des Netzwerks eindeutig sein und darf bis zu 15 Zeichen lang sein, wobei die folgenden Zeichen nicht verwendet werden dürfen:

« / \ [ ] ; : | < > + = , ? * und das Leerzeichen

- *domain*: Hierüber wird die Standarddomäne für diesen Rechner definiert. Findet ein Anmeldevorgang mit Validierung in der Domäne statt, ohne daß ein bestimmte Domäne angegeben wird, wird die Standarddomäne verwendet. Auch dieser Name muß absolut eindeutig im Netzwerk sein und darf ebenfalls die folgenden Zeichen nicht enthalten:

« / \ [ ] ; : | < > + = , ? * und das Leerzeichen

- *othdomains*: Hier können bis zu vier Domänen angegeben werden, an denen der Rechner teilnehmen kann. Die Domänen müssen durch das Zeichen »;« getrennt werden. Der Rechner erhält dann alle Nachrichten und Fehlermeldungen, die innerhalb dieser Domänen geschickt werden. Soll der Rechner an einer Windows-NT-Domäne oder einer Windows-95-Arbeitsgruppe teilnehmen, ist hier die Windows-NT-Domäne bzw. die Windows-95-Arbeitsgruppe einzutragen. Damit sind auch die Rechner dieser Domänen beziehungsweise Arbeitsgruppen im LAN-Requester-Ressourcen-Browser zu sehen. Es können dadurch die Ressourcen von bis zu fünf Domänen automatisch im Browser angezeigt werden.
- *wrknets*: Unter diesem Parameter werden die zu verwendenden logischen Netzwerke in einer durch Kommata getrennten Liste aufgeführt, so wie sie in der Networks-Sektion aufgeführt sind. Die Vorgabe ist »net1«. Wird der LoopBack-Treiber verwendet, muß »net1« der erste Parameter in dieser durch Kommata getrennten Liste sein.
- *wrkservices*: Hier werden die automatisch zu startenden Dienste eingetragen. Standardmäßig werden die Dienste *Messenger* und *Peer* gestartet. Die Zuordnung der dort angegebenen Namen zu den ausführbaren Programmen findet sich in der Services-Sektion wieder. Nimmt der Rechner im Normalfall nicht an einem Peer-Netzwerk teil, kann das automatische Starten unterbunden werden, indem der Peer Dienst als Parameter gelöscht wird. Werden auch die Nachrichtendienste nicht verwendet, können auch sie gelöscht werden. Dies führt allerdings dazu, daß nach der Anmeldung in der Domäne eine Fehlermeldung angezeigt wird, daß der Benutzer keine Nachrichten empfangen kann. Diese Fehlermeldung kann jedoch unterdrückt werden, indem in den *wrkheusristics* das 38. Bit auf den Wert 2 gesetzt wird. Achtung, dadurch wird keinerlei Fehlermeldung der Domäne mehr angezeigt! Ferner könnte noch das 36. Bit der *wrkheusristics* von Interesse sein, wenn ein automatischer Zeitabgleich zwischen dem Domänencontroller und dem Requester nicht gewünscht wird. Die weiteren Parameter sind in der Hilfe »LAN Requester Dienste Handbuch« beschrieben, bedürfen

jedoch nur in Ausnahmefällen der Konfiguration. Eine Reihe von Parametern existiert auch in der Peer-Sektion, in der sie für die Peer-Dienste konfiguriert werden können.

– *charcount*: Um bei einem Zugriff auf eine serielle Einheit nicht jedes Zeichen gesondert über das Netz zu schicken, werden die ausgehenden Zeichen bis zu der in »charcount=« angegebenen Anzahl gepuffert, um dann gesammelt über das Netz zu gehen. Sollen viele Daten an freigegebene serielle Einheiten gesendet und zugleich der Netzverkehr gesenkt werden, kann dieser Parameter erhöht werden. Der Wertebereich reicht von 0 bis 65535 (Vorgabe: 16).

– *chartime*: Der für »chartime=« angegebene Wert legt fest, wie lange, in Millisekunden, ein Requester Daten für eine serielle Einheit sammelt, bevor sie über das Netz geschickt werden. Eine Erhöhung dieses Wertes bewirkt eine Reduzierung der Netzbelastung. Der Wertebereich reicht von -1 bis 65535 (Vorgabe: 250). Bei einem Wert von -1 wartet der Requester so lange, bis die in »charcount=« angegebene Anzahl an Zeichen zur Übertragung ansteht.

– *charwait*: Dieser Parameter legt fest, wie lange, in Sekunden, der Requester auf die Verfügbarkeit einer seriellen Einheit wartet. Greifen viele Benutzer auf eine freigegebene serielle Einheit zu, kann es notwendig werden, daß dieser Wert erhöht wird. Der Wertebereich reicht von 0 bis 65535 (Vorgabe: 3600).

– *keepconn*: Greift eine Applikation direkt über einen UNC- (Universal Naming Convention) Namen auf eine Ressource zu (das bedeutet, daß zuvor kein NET USE abgesetzt wurde), legt dieser Parameter fest, wie viele Sekunden diese Verbindung inaktiv sein darf, bevor sie getrennt wird. Der Wertebereich reicht von 1 bis 65535 (Vorgabe: 600).

– *keepsearch*: Dieser Parameter legt fest, wie lange eine inaktive Dateisuche aufrecht erhalten wird. Der Wertebereich reicht von 1 bis 65535 (Vorgabe: 600).

– *maxcmds*: Dieser Parameter spezifiziert die maximale Anzahl an NetBios-Befehlen, die der Requester gleichzeitig an den Netzwerkadapter senden kann. Um die Performance zu erhöhen, wenn viele Applikationen gleichzeitig auf das Netzwerk zugreifen, kann dieser Wert erhöht werden. Dies geht jedoch zu Lasten des Speicherbedarfs. Idealerweise ist der Wert für *maxcmds* das 1,6-fache des Wertes für *maxthreads*. Der Wertebereich reicht von 5 bis 255 (Vorgabe: 16).

– *maxerrorlog*: Die maximale Größe des Fehlerprotokolls wird über diesen Parameter festgelegt. Der angegebene Wert bezeichnet die Größe des Fehlerprotokolls in Kilobyte. Der Wertebereich reicht von 2 bis zu der vorhandenen Festplattenkapazität (Vorgabe: 100 Kbyte).

– *maxthreads*: Über den Parameter *maxthreads* wird festgelegt, wieviele *Threads* dem Requester gleichzeitig zur Verfügung stehen, um Netzwerkanfragen abzuarbeiten. Die effektiv zur Verfügung stehende Anzahl an Threads ist abhängig von den Parametern *maxcmds*, *maxthreads* und *numworkbuf*. Greifen mehrere Applikationen gleichzeitig auf Netzwerkressourcen zu, können zur Performancesteigerung alle drei Parameter erhöht werden. Die Parameter *maxcmds* und *numworkbuf* sollten jedoch immer größer sein als *maxthreads*. Der Wertebereich reicht von 10 bis 254 (Vorgabe: 10).

– *maxwrkcache*: Dieser Parameter legt die maximale Größe (in Kilobyte) der großen Transferpuffer des Requesters fest. Diese werden für die Übertragung großer Dateien verwendet, sofern sie sowohl auf dem (Peer-)Server als auch auf dem Requester zur Verfügung stehen. Der angegebene Wert sollte ein vielfaches von 64 sein. Um eine optimale Nutzung der großen Transferpuffer zu gewährleisten, sollte der Wert auf allen Rechnern innerhalb der Domäne (vor allen Dingen zwischen Servern und Requestern) identisch sein. Nur dann können die großen Transferpuffer auch bis zum letzten Byte ausgenutzt werden. Der Wertebereich reicht von 0 bis 640 (Vorgabe: 64).

– *numalerts*: Dieser Parameter legt die Anzahl an Programmen fest, die auf eine Fehlerbedingung warten können. Eine Erhöhung dieses Werts ist im Normalfall nicht notwendig. Zudem bringt eine Erhöhung dieses Werts einen erhöhten Speicherbedarf mit sich. Der Wertebereich reicht von 5 bis 200 (Vorgabe: 12).

– *numcharbuf*: Werden Daten an eine freigegebene serielle Einheit (z.B. Modem) übertragen, werden dafür spezielle Puffer verwendet. Über *numcharbuf* wird festgelegt, wieviele solche Character Pipes und Device Puffer zur Verfügung gestellt werden sollen. Der Wertebereich reicht von 0 bis 15 (Vorgabe: 10).

– *numservices*: Zur Verwaltung der zur Verfügung stehenden Dienste wird eine Servicetabelle angelegt. Dieser Parameter legt fest, wie viele Dienste in diese Tabelle aufgenommen werden können. Der verwendete Wert sollte größer oder mindestens gleich der Anzahl der in der Services-Sektion eingetragenen Dienste sein. Der Wertebereich reicht von 4 bis 256 (Vorgabe: 4).

– *numworkbuf*: Dieser Parameter legt die Anzahl der zur Verfügung stehenden (»normalen«) Puffer fest. Diese werden sowohl für den Verbindungsaufbau (Aufbau des Messageblocks) vor einer Datenübertragung verwendet als auch, wenn keine großen Transferpuffer mehr zur Verfügung stehen. Dann nämlich einigen sich Requester und Server auf die Übertragung der Daten unter Verwendung der »normalen« Puffer. Eine Änderung dieses Wertes bringt in der Regel kaum einen Performancegewinn. Der Wertebereich reicht von 3 bis 50 (Vorgabe: 15).

– *numdgrambuf*: Zum Versenden von sogenannten Broadcasts werden NetBios-Datagramme verwendet. Beispiele hierfür wären eine domänenweite Nachricht oder die Bekanntgabe der eigenen Existenz eines Rechners beim Hochfahren. Bei einer Kommunikation zwischen Rechnern unter Verwendung von Datagrammen kann eine ordnungsgemäße Zustellung (richtige Reihenfolge bzw. die Zustellung überhaupt) nicht garantiert werden. Hier muß die Applikation selbst für Datenintegrität sorgen. Der Parameter *numgradbuf*= legt nun fest, wieviele Puffer zur Aufnahme solcher Datagramme zur Verfügung gestellt werden sollen. Der Wertebereich reicht von 8 bis 112 (Vorgabe: 14).

– *othdomains*: Siehe oben

– *printbuftime*: Dieser Wert legt fest, nach welcher Zeit (in Sekunden) ein Druckauftrag von einer DOS-Applikation als beendet angesehen wird, sofern kein »End of Job«-Signal gesendet wurde. Wird bei Drucken aus einer DOS-Applikation der Ausdruck unvollständig oder verstümmelt ausgedruckt, kann dieser Wert erhöht werden, damit der Datenstrom von der DOS-Applikation zum Drucker nicht unterbrochen wird. Der Wertebereich reicht von 0 bis 65535 (Vorgabe: 90).

– *sesstimeout*: Eine inaktive Verbindung von einer anderen Peer-Workstation wird nach der in *sesstimeout=* angegebenen Anzahl Sekunden automatisch getrennt. Kommt es hin und wieder zu unmotivierten Unterbrechungen einer Verbindung zwischen zwei Rechnern, ist dieser Wert zu erhöhen. Der Wertebereich reicht von 10 bis 65535 (Vorgabe: 45).

– *sizcharbuf*: Dieser Parameter legt die Größe (in Byte) der Read-Ahead-Puffer für zeichenorientierte Einheiten (z.B. Modem) fest. Werden große Datenmengen über das Netzwerk an oder von seriellen Einheiten übertragen, bringt eine Erhöhung dieses Wertes einen erhöhten Datendurchsatz. Der Wertebereich reicht von 64 bis 4096 (Vorgabe: 512).

– *sizerror*: Über *sizerror=* wird die Größe des zur Verfügung gestellten internen Puffers für Netzwerkfehler (in Byte) festgelegt. Der Wertebereich reicht von 256 bis 4096 (Vorgabe: 1024).

– *sizworkbuf*: Dieser Parameter legt die Größe des »normalen« Puffers (im Gegensatz zu den großen Transferpuffern) fest. Dieser Wert sollte ein vielfaches von 512 sein. Zudem sollte dieser Wert auf allen Rechner im Netzwerk identisch sein. Aus diesem Grund kann nicht empfohlen werden, Änderungen an diesem Wert vorzunehmen. Der Wertebereich reicht von 1024 bis 16384 (Vorgabe: 4096).

– *useallmem*: Mit diesem Parameter wird festgelegt, ob die Netzwerkkarte in der Lage ist, Speicher oberhalb 16 Mbyte zu adressieren. Ein Wert von »no« zwingt die Netzwerkkarte, Speicher nur unterhalb von 16 Mbyte zu verwenden. Beim Wert »yes« kann sowohl Speicher oberhalb als auch unterhalb der 16-Mbyte-Grenze verwendet werden.

Die *wrkheuristics=* in der Requester-Sektion beschreiben eine Reihe weiterer Parameter zur Feinabstimmung des Requesters. Sie sind im einzelnen in der »ibmlan.ini« beschrieben. Eine Änderung sollte nur dann vorgenommen werden, wenn eindeutig geklärt wurde, welche Auswirkungen die beabsichtigte Änderung mit sich bringt.

```
[requester]
; The following parameters generally do not need to be
; changed by the user.
charcount = 16
chartime = 250
charwait = 3600
keepconn = 600
keepsearch = 600
maxcmds = 16
maxerrorlog = 100
maxthreads = 10
maxwrkcache = 64
numalerts = 12
numcharbuf = 10
numservices = 4
numworkbuf = 15
numdgrambuf = 14
othdomains = MUC_NT;Aibling
printbuftime = 90
```

```
sesstimeout = 45
sizcharbuf = 512
sizerror = 1024
sizworkbuf = 4096
useallmem = no
; The wrkheuristics parameter sets a variety of requester fine-tuning
; options.  Brief descriptions are provided here for each digit.  More
; complete descriptions, along with explanations of relationships between
; digits can be found in the OS/2 Peer Technical Reference.
;
; When not otherwise specified, a value of 0 means off (inactive) and 1 means
; on (active).
;
; 0   opportunistic locking of files (default is 1)
; 1   performance optimization for batch (.CMD) files (default is 1)
; 2   asynchronous unlock and asynchronous write-unlock
;        0 = never         2 = only on a LAN Server virtual circuit
;        1 = always (default)
; 3   asynchronous close and asynchronous write-close
;        0 = never         2 = only on a LAN Server virtual circuit
;        1 = always (default)
; 4   buffer all named pipes and serial devices (default is 1)
; 5   combined read-lock and write-unlock
;        0 = never         2 = only on a LAN Server virtual circuit
;        1 = always (default)
; 6   open and read optimization (default is 1)
; 7   reserved
; 8   use of the chain-send NETBIOS NCB
;        0 = never
;        1 = when a server's buffer is larger than the workstations buffer
;        2 = always (default)
buffer small read and write requests until the buffer is full
;        0 = never         2 = only on a LAN Server virtual circuit
;        1 = always (default)
; 10  buffer mode
;        0 = always read buffer size amount of data if the request is smaller
;            than the buffer size (sizworkbuf) and data is being read
;            sequentially
;        1 = use full buffer if file is open for reading and writing
;        2 = use full buffer if reading and writing sequentially
;        3 = buffer all requests smaller than the buffer size (if hits occur)
;            (default)
; 11  RAW read and RAW write SMB protocols (default is 1)
RAW read-ahead buffer (default is 1)
; 13  RAW write-behind buffer (default is 1)
; 14  read multiplexing SMB protocols (default is 1)
; 15  write multiplexing SMB protocols (default is 1)
; 16  use of big buffers for large core (non-RAW) reads (default is 1)
; 17  same-size read-ahead or read-to-sector boundary (default is 1)
```

```
; 18  same-size small record write-behind or write-to-sector boundary
;     (default is 0)
; 19  reserved and must be 0
; 20  flush pipes and devices on a DosBufReset or DosClose operation
;        0 = Flush only files and devices opened by the caller.  Spin until
;            flushed.  Wait for confirmation before processing with other
;            tasks (default)
;        1 = Flush only files and devices opened by the caller.  Flush only
;            once. Do not wait for confirmation.
;        2 = Flush all files and all input and output of short-term pipes
;            and devices. Spin until flushed.
;        3 = Flush all files and all input and output of short-term pipes
;            and devices. Flush only once.
;        4 = Flush all files and all input and output of all pipes and devi-
ces.
;            Spin until flushed.
;        5 = Flush all files and all input and output of all pipes and devi-
ces.
;            Flush only once.
; 21  LAN Server encryption of passwords (default is 1)
; 22  control log entries for multiple occurrences of an error
;        0 = log all occurrences (default)
;        1-9 = limit occurrences that are logged (1-9 define size of table
;            that is used to track errors)
; 23  buffer all files opened with deny-write sharing mode (default is 1)
; 24  buffer all files opened with the read-only attribute set (default is 1)
; 25  read ahead when opening for execution (default is 1)
; 26  handle the interrupt (ctrl+c) key
;        0 = allow no interrupts
;        1 = allow interrupts only on long-term operations
;        2 = always allow interrupts (default)
; 27  force correct open mode when creating files on a core server (reserved
;        for DBCS users) (default is 1)
; 28  NETBIOS NoAck mode
;        0 = NoAck is disabled.
;        1 = NoAck is set on send only. (default)
;        2 = NoAck is set on receive only.
;        3 = NoAck is set on send and receive.
; 29  send data along with SMB write-block RAW requests (default is 1)
; 30  send a message to the screen when the requester logs an error
;        0 = never
;        1 = on write-fault errors only (no time out) (default)
;        2 = on write-fault and internal errors only (no time out)
;        3 = on all errors (no time out)
;        4 = reserved
;        5 = on write-fault errors only (time out)
;        6 = on write-fault and internal errors only (time out)
;        7 = on all errors (time out)
;        8 = on all errors (time out) and pop up an error if a session
```

```
;            to a server is abnormally disconnected
;        9 = on all errors (time out) and generate an INT24 (critical
;            error) to applications running under DOS emulation if a
;            network hard error is encountered
; 31  reserved
; 32  behavior of DosBufReset on a redirected file (not pipes or devices).
;     When the call to the API returns, the data in the buffer has been
;     handled as follows:
;        0 = Changed data in the buffer was sent from the requester to the
;            server. The server has written the data to disk.
;        1 = Changed data in the buffer was sent from the requester to the
;            server. The server has not yet written the data to disk.
;        2 = DosBufReset was ignored. (default)
; 33  time interval for performing logon validation from the domain
;     controller
;        0 = 5 seconds                 4 = 60 seconds      8 = 8 minutes
;        1 = 15 seconds (default)      5 = 90 seconds      9 = 15 minutes
;        2 = 30 seconds                6 = 2 minutes
;        3 = 45 seconds                7 = 4 minutes
; 34  date validation
;        0 = PCLP date format (default)
;        1 = MSNET date format
;        2 = assume date is correct
; 35  free disk space reported to DOS and Windows applications
;        0 = return true value (default)
;        1 = return a value less than two gigabytes
; 36  time and date synchronization with the domain controller at logon
;     (default is 1)
; 37  typ of verification for the LAN Server logon
;        0 = no verification
;        1 = verify against local NET.ACC (default)
;        2 = verify against domain NET.ACC
; 38  display warning messages for the LAN Server logon
;        0 = do not display warning messages
;        1 = display all warning messages (default)
;        2 = do not display LAN Server specific warning messages
;        3 = display all warning messages, including password expiration date
; 39  buffer all files opened in compatibility mode
;        0 = buffer only files opened for read access in compatibility mode
;        1 = buffer all files opened in compatibility mode (default)
; 40  allow a user to logon multiple times in the same domain (default is 1)
;
; The next lines help you to locate bits in the wrkheuristics entry.
;                           1         2         3         4
;                 012345678901234567890123456789012345678901
wrkheuristics = 111111112131111111000101112011122100111110
wrknets = NET1
wrkservices = MESSENGER,PEER
Computername = FS_THINK
Domäne = LS4DESKTOP
```

Messenger-Sektion

Die Messenger-Sektion beinhaltet nur zwei Parameter. Der erste Parameter *logfile* legt den Namen der Logdatei für den Nachrichtendienst fest. Die dort angegebene Datei wird im Verzeichnis »\ibmlan\logs« erstellt. Der Vorgabename ist »messages.log«.

Der zweite Parameter *sizmessbuf* gibt an, wie groß der Puffer in Byte für die eingehenden Nachrichten sein kann. Werden viele und große Nachrichten empfangen, ist dieser Wert zu erhöhen. Dies erhöht jedoch den Speicherbedarf.

```
[messenger]
logfile = messages.log
sizmessbuf = 4096
```

Die Peer-Sektion

Die Peer-Sektion ist für die Konfiguration der Peer-Dienste verantwortlich und übernimmt hier im wesentlichen die Konfiguration der Serverkomponente innerhalb des Peer-Dienstes. Folgende Parameter stehen für die Konfiguration zur Verfügung:

- *auditing*: Dieser Parameter legt fest, welche Ereignisse aufgezeichnet werden sollen, falls Auditing für die Peer-Dienste aktiviert wurde. Hierzu müssen die Peer-Dienste unter Angabe des optionalen Parameters »/AUDITING« gestartet werden (NET START PEER /AUDITING:YES). Damit werden alle Ereignisse, die in der »ibmlan.ini« für das Schlüsselwort *auditing=* gesetzt wurden, aufgezeichnet. Es können beliebig viele Ereignisse als Liste angegeben werden. Es darf jedoch keine Ereignis angegeben werden, das einen Unterpunkt eines anderen Ereignisses darstellt (z.B. NETLOGON und BADNETLOGON, da BADNETLOGON ein Unterpunkt der Aktion NETLOGON darstellt). Folgende Ereignisse können mittels Auditing überwacht werden:
- *Logon*: Mittels »Logon« wird überwacht, wann ein Benutzer einen der Peer-Dienste startet oder beendet. Logon beinhaltet die Ereignisse »NetLogon« und »SessLogon«.
- *LogonLimit*: Erstellt einen Eintrag, sobald ein Benutzer seinen definierten Anmeldezeitraum überschreitet.
- *NetLogon*: Erstellt einen Eintrag, wenn sich ein Benutzer anmeldet. *NetLogon* schließt die Ereignisse *BadNetLogon* sowie *GoodNetLogon* mit ein.
  - *BadNetLogon*: Ein Eintrag wird erstellt, wenn ein Anmeldeversuch fehlgeschlagen ist.
  - *GoodNetLogon*: Ein Eintrag wird erstellt, wenn ein Anmeldeversuch erfolgreich war.
- *Permissions*: Erstellt einen Eintrag, wenn ein Benutzer Änderungen an den Zugriffsberechtigungen einer Ressource durchführt.
- *Ressource*: Erstellt einen Eintrag, wenn ein Benutzer auf eine Ressource zugreift. Die effektiv aufgezeichneten Ereignisse hängen von den in der Ressource definierten Auditing-Optionen ab. Dies ist zugleich der Vorgabewert für das Schlüsselwort *auditing=*.
- *Service*: Erstellt einen Eintrag, wenn ein Benutzer einen der Peer-Dienste startet oder beendet.

– *SessLogon*: Erstellt einen Eintrag, wenn eine Verbindung zum Peer-Rechner aufgebaut oder beendet wird. *SessLogon* beinhaltet die Ereignisse *BadSessLogon* sowie *GoodSessLogon*.

  – *BadSessLogon*: Erstellt einen Eintrag, wenn eine Verbindung nicht aufgebaut werden konnte.

  – *GoodSessLogon*: Erstellt einen Eintrag, wenn eine Verbindung erfolgreich aufgebaut wurde.

– *Use*: Erstellt einen Eintrag, wenn eine freigegebene Ressource aktiv benutzt wird. *Use* beinhaltet die Ereignisse *BadUse* und *GoodUse*.

  – *BadUse*: Erstellt einen Eintrag, wenn eine Benutzung einer freigegebenen Ressource fehlgeschlagen ist.

  – *GoodUse*: Erstellt einen Eintrag, wenn eine Benutzung einer freigegebenen Ressource erfolgreich war. Wurde die Ressource für einen unbegrenzten Zugriff freigegeben, wird kein Eintrag erstellt.

– *UserList*: Erstellt einen Eintrag, wenn ein Benutzer Änderungen an der Benutzerdatenbank vornimmt (z.B. Benutzer anlegen/löschen oder Benutzerprofil ändern).

– *guestacct*: Hier kann die UserID angegeben werden, die für einen Gastzugriff auf eine freigegebene Ressource verwendet werden soll. Will ein Benutzer auf eine Ressource zugreifen, ohne daß er über eine Gruppenzugehörigkeit oder als Benutzer selbst Rechte für die Ressource besitzt, kann er den hier angegebenen Gastaccount benutzen, sofern er für diesen Rechte vorhanden sind. Ist dies der Fall, wird der Zugriff gemäß den Rechten des Gastaccounts gewährt. Die Vorgabe für diesen Account ist »GUEST«.

– *autodisconnect*: Dieser Wert gibt die Zeit in Minuten an, nach der inaktive Verbindungen getrennt werden. Eine Zahl zwischen 1 und 32767 entspricht der Wartezeit, die verstreichen muß, bevor die Verbindung getrennt wird. Beim Wert -1 (der Vorgabe) erfolgt keine automatische Trennung von inaktiven Sitzungen. Der Wert 0 sollte nicht verwendet werden, da dadurch Fehler generiert werden. Wird beispielsweise über den Wert für *maxusers=* der gleichzeitige Zugriff auf den Rechner für fünf Benutzer ermöglicht, kann über *autodisconnect=* sichergestellt werden, daß eine inaktive Sitzung nicht den Zugriff eines weiteren Benutzers blockiert.

– *forwardauth*: Mit diesem Schlüsselwort wird kontrolliert, ob *Forwarded Authentication* für die Validierung eines Benutzers verwendet werden soll. Wird *Forwarded Authentication* mittels »no« ausgeschaltet (die Vorgabe), muß ein Benutzer auf dem lokalen Rechner einen Benutzeraccount besitzen, um auf Ressourcen zugreifen zu können, die auf Benutzerebene freigegeben wurden. Ferner muß sichergestellt sein, daß das Passwort für die Anmeldung auf dem Peer-Rechner identisch ist mit dem Paßwort für den eigenen Rechner oder dem Paßwort auf anderen Peer-Rechnern. Wird *Forwarded Authentication* verwendet, übergibt der Peer-Rechner, der die Ressource bereitstellt, den Logon-Namen sowie das Paßwort an den Domaincontroller für die Validierung. Besitzt die Logon-Namen/Passwort-Kombination einen gültigen Account in der Domäne, wird der Zugriff auf die Ressource des Peer-Rechners gemäß den Rechten des in der Domäne definierten Benutzers definiert. Befindet sich

ein Domaincontroller im verwendeten Netzwerk, sollte *Forwarded Authentication* auf allen Peer-Rechnern verwendet werden. Dadurch entfällt die Notwendigkeit, das Paßwort auf jedem einzelnen Rechnern zu synchronisieren. Es muß lediglich das Paßwort auf dem eigenen Rechner sowie das Paßwort in der Domäne abgeglichen werden.

- *maxauditlog*: Dieser Wert legt die maximale Größe der Audit-Protokolldatei in Kilobyte fest. Wird die hier angegebene Dateigröße erreicht, werden keine weiteren Einträge in die Protokolldatei geschrieben, bis Einträge gelöscht werden. Der Wertebereich reicht von 0 bis 65535 (Vorgabe: 100).
- *maxchdevjob*: Mit diesem Parameter wird angegeben, wieviele Anfragen an freigegebene serielle Ressourcen von einem Peer-Rechner in der Summe akzeptiert werden können. Der Wertebereich reicht von 0 bis 65535 (Vorgabe: 2).
- *maxchdevq*: Der Wert für *maxchdevq=* legt die maximale Anzahl an Warteschlangen für serielle Einheiten fest. Der Wertebereich reicht von 0 bis 65535 (Vorgabe: 1).
- *maxchdevs*: Über diesen Wert wird festgelegt, wieviele serielle Einheiten freigegeben werden können. Dieser Wert muß angepaßt werden, wenn mehr als eine serielle Einheit freigegeben werden soll. Der Wertebereich reicht von 0 bis 16 (Vorgabe: 1).
- *maxconnections*: Dieser Parameter legt die maximale Anzahl an Verbindungen zum Peer-Rechner fest. Dieser Wert muß mindestens so groß sein wie der Wert für *maxusers=*, damit die dort definierte Anzahl an Benutzern auch eine Verbindung zu dem Peer-Rechner aufbauen kann. Jeder Zugriff auf eine Ressource mittels NET USE stellt eine Verbindung dar. Der Wertebereich reicht von 1 bis 200 (Vorgabe: 128).
- *maxlocks*: Greifen DOS-Rechner auf den OS/2-Peer-Rechner zu, wird über den Wert dieses Schlüsselwortes festgelegt, wieviele Dateisperren möglich sind. Der Wertebereich reicht von 1 bis 8000 (Vorgabe: 64).
- *maxopens*: Der Wert für *maxopens=* legt fest, wieviele Dateien, Pipes oder serielle Einheiten auf dem Rechner gleichzeitig geöffnet sein können. Greifen beispielsweise drei Benutzer auf eine Datei zu, muß der Wert mindestens 3 sein. Greift umgekehrt ein Benutzer auf drei Dateien zu, muß dieser Wert ebenfalls mindestens 3 sein. Der Wertebereich reicht von 1 bis 8000 (Vorgabe: 160).
- *maxsearches*: Führt ein Benutzer eine Dateisuche auf eine freigegebene Ressource aus (z.B. DIR *.* /S), legt dieser Parameter fest, wie viele derartige Dateisuchen gleichzeitig auf dem Peer-Rechner ausgeführt werden können. Dieser Wert steht im direkten Zusammenhang des 7. Flags der *srvheuristics*, in dem festgelegt wird, ob automatisch mehr Speicher für die Dateisuche angefordert werden soll (Vorgabe). Der Wertebereich reicht von 1 bis 1927 (Vorgabe: 150).
- *maxsessopens*: Der Wert dieses Schlüsselwortes legt fest, wieviele Dateien, Einheiten oder Pipes von einem Requester gleichzeitig geöffnet werden dürfen. Die effektiv für einen Requester zur Verfügung stehende Anzahl liegt jedoch darunter, da auch für die interne Verarbeitung Dateien, Pipes oder Einheiten geöffnet werden. Der Wertebereich reicht von 1 bis 8000 (Vorgabe: 120). Der angegebene Wert muß unter dem Wert von *maxopens=* liegen.

– *maxsessreqs*: Über diesen Eintrag wird festgelegt, wieviele Anfragen an Ressourcen eines Peer-Rechners von einem Requester gleichzeitig vorliegen dürfen. Der Wertebereich reicht von 1 bis 65535 (Vorgabe: 50).

– *maxsessvcs*: Der Wert für dieses Schlüsselwort muß auf 1 gesetzt sein. Damit wird die maximale Anzahl sogenannter »virtual circuits«, die der Peer-Rechner von einem Requester akzeptieren kann, festgelegt.

– *maxshares*: Dieser Parameter legt fest, wieviele Ressourcen von dem Peer-Rechner gleichzeitig freigegeben werden können. Soll ein anderer Benutzer auf drei verschiedene Freigaben zugreifen können, ist dieser Wert auf mindestens 3 zu erhöhen. Greifen umgekehrt mehrere Benutzer auf ein- und dieselbe Freigabe zu, ist der Mindestwert von 2 ausreichend. Der Wertebereich reicht von 2 bis 1000 (Vorgabe: 64). Überprüft man die Anzahl an möglichen Freigaben mittels NET CONFIG PEER, ist zu berücksichtigen, daß zusätzlich zu der hier konfigurierten Anzahl noch die administrativen Freigaben wie IPC$, ADMIN$, C$ usw. hinzugerechnet werden.

– *maxusers*: Der Wert für *maxusers=* legt fest, wieviele Benutzer gleichzeitig auf den Peer-Rechner zugreifen können. Hierbei zählen nicht die Anzahl der Verbindungen (durch NET-USE-Befehle), sondern die effektive Anzahl an Benutzer (NetBios-Verbindungen). Greift ein Benutzer auf mehrere Freigaben zu, ist nur eine NetBios-Verbindung in Gebrauch. Greifen dagegen drei Benutzer auf eine Ressource zu, werden bereits drei NetBios-Verbindungen benötigt. Der Wertebereich reicht von 1 bis 1000 (Vorgabe: 32). Für das Maximum von 1000 Benutzern müssen jedoch vier Netzwerkkarten konfiguriert sein (net1 bis net4 in der Net-Work-Sektion der IBMLAN.INI), da pro Netzwerkkarte maximal 250 Benutzer möglich sind.

– *numbigbuf*: Dieser Parameter legt die Anzahl an großen Transferpuffern fest. Diese werden für die Übertragung großer Dateien wie z.B. bei einem Programmstart oder beim Kopieren großer Dateien verwendet. Werden Dateien sehr langsam über das Netzwerk kopiert, ist davon auszugehen, daß die Anzahl der großen Transferpuffer nicht ausreicht. In direktem Zusammenhang mit den großen Transferpuffern stehen das 13., 17. und 18. Flag der *srvheuristics*. Diese legen das Verhalten bezüglich der Speicheranforderung fest. Der Wertebereich reicht von 0 bis 80 (Vorgabe: 6). Der konfigurierte Wert sollte die Anzahl der Benutzer, die große Dateiübertragungen benötigen, mal 2 sein. Bei fünf Benutzern ergibt sich also der Wert 10.

– *numfiletasks*: Über diese Wert wird festgelegt, wieviele Tasks (Prozesse) für das Dateihandling zur Verfügung gestellt werden sollen. Da diese Prozesse multithreaded sind, stehen pro Prozess 48 Threads zur Verfügung. Der Wertebereich reicht von 1 bis 8 (Vorgabe: 1). Reichen die zur Verfügung gestellten Tasks nicht aus, wird ein Fehler in der Fehlerprotokolldatei protokolliert. In diesem Fall sollte der Wert erhöht werden.

– *numreqbuf*: Dieser Parameter legt fest, wie viele Request-Puffer der Größe *sizereqbuf* zur Verfügung gestellt werden sollen. Diese Request-Puffer werden für die Übertragung kleinerer Dateien beziehungsweise für den Verbindungsaufbau bei der Übertragung großer Dateien verwendet. Stehen für die Übertragung großer Dateien keine großen Transferpuffer mehr zur Verfügung, werden ebenfalls diese Request-Puffer verwendet. Der Wertebereich reicht von 5

bis 2000 (Vorgabe: 32). Für jeden Benutzer, der auf den Peer-Rechner zugreift, sollten zwei Request-Puffer eingetragen werden (plus zwei für jede verwendete Pipe).

– *sizreqbuf*: Über *sizereqbuf=* wird die Größe der Request-Puffer in Byte festgelegt. Dieser Wert sollte auf allen Peer-Rechnern gleich groß sein und der Größe von *sizeworkbuf=* auf reinen Requester-Maschienen entsprechen, um eine optimale Ausnutzung der Systemressourcen zu gewährleisten. Der Wertebereich reicht von 1024 bis 32768 (Vorgabe: 4096).

– *srvanndelta*: Dieser Wert legt fest, in welchem Zeitrahmen (in Millisekunden) der Rechner die Bekanntgabe seiner Existenz im Netzwerk variieren kann. Der Wertebereich reicht von 0 bis 65535 (Vorgabe: 3000).

– *srvannounce*: Dieser Wert legt fest, in welchen Zeitabständen (in Sekunden) dieser Rechner seine Existenz im Netzwerk bekanntgibt. Der Wertebereich reicht von 0 bis 65535 (Vorgabe: 180).

– *srvhidden*: Mit diesem Parameter wird die Bekanntgabe der eigenen Existenz im Netzwerk unterdrückt. Mögliche Werte sind »no« (die Vorgabe) und »yes«. Bei »yes« wird das 20. Flag der *srvheuristics* außer Kraft gesetzt.

– *srvcomment*: Dieses Schlüsselwort ist standardmäßig nicht in der »ibmlan.ini« vorhanden und enthält einen Kommentar des Rechners.

– *srvheuristics*: Die *srvhesuristics=* in der Peer-Sektion beschreiben eine Reihe weiterer Parameter zur Feinabstimmung der Peer-Dienste. Sie sind im einzelnen in der »ibmlan.ini« beschrieben. Eine Änderung sollte nur dann vorgenommen werden, wenn eindeutig geklärt wurde, welche Auswirkungen die beabsichtigte Änderung mit sich bringt.

– *srvservices*: Hier wird angegeben, welche Dienste beim Starten der Peer-Dienste automatisch mit gestartet werden sollen. Die Dienste werden als eine durch Kommata getrennte Liste angegeben. Die verwendeten Namen der Dienste beziehen sich auf die Bezeichnungen der Dienste in der Sektion »Services«.

– *srvnets*: Unter diesem Schlüsselwort werden alle zu verwendenden Netzwerke aufgelistet. Die Bezeichnung »Netzwerk« bezieht sich in diesem Fall auf alle in der Network-Sektion der »ibmlan.ini« aufgelisteten Netzwerke net1 bis netx. Die Vorgabe ist net1.

```
[peer]
 auditing = resource
; The following parameters generally do not need to be
; changed by the user. NOTE : srvnets= is represented in
; the server info struct as a 16-bit lan mask. Srvnet names
; are converted to indexes within [networks] for the named nets.
 guestacct = guest
 autodisconnect = -1
 forwardauth = no
 maxauditlog = 100
 maxchdevjob = 6
 maxchdevq = 2
 maxchdevs = 2
```

```
 maxconnections = 128
 maxlocks = 64
 maxopens = 160
 maxsearches = 150
 maxsessopens = 120
 maxsessreqs = 25
 maxsessvcs = 1
 maxshares = 64
 maxusers = 32
 numbigbuf = 6
 numfiletasks = 1
 numreqbuf = 48
 sizreqbuf = 4096
 srvanndelta = 3000
 srvannounce = 180
 srvhidden = yes
; The srvheuristics parameter sets a variety of server fine-tuning
; options.  Brief descriptions are provided here for each digit.  More
; complete descriptions, along with explanations of relationships between
; digits can be found in the OS/2 Peer Technical Reference.
;
; When not otherwise specified, a value of 0 means off (inactive) and 1 means
; on (active).
;
; 0   opportunistic locking of files (default is 1)
; 1   read-ahead when the requester is performing sequential access
;       0 = do not use read-ahead
;       1 = use single read-ahead thread (default)
;       2 = use asynchronous read-ahead thread
; 2   write-behind (default is 1)
; 3   use of the chain-send NETBIOS NCB (default is 1)
; 4   check all incoming SMBs for correct format (default is 0)
; 5   support FCB opens (default is 1)
; 6   set priority of server (default is 4)
;       0-9 = 0 is highest priority, 9 is lowest priority
; 7   automatically allocate additional memory for directory searches
;         (default is 1)
; 8   write records to the audit log only when the scavenger wakes up
;         (default is 1)
; 9   do full buffering when a file is opened with deny-write sharing mode
;         (default is 1)
; 10  set the interval for the scavenger to wake up
;       0 = 5 seconds                  4 = 25 seconds      8 = 45 seconds
;       1 = 10 seconds (default)       5 = 30 seconds      9 = 50 seconds
;       2 = 15 seconds                 6 = 35 seconds
;       3 = 20 seconds                 7 = 40 seconds
; 11  allow compatibility-mode opens of certain typs of files by translating
;     them to sharing mode opens with deny-none
;       0 = Always use compatibility-mode opens.
```

```
;       1 = Use deny-none sharing mode if read-only access to .EXE or .COM
;           files.
;       2 = Use deny-none sharing mode if read-only access to .EXE or .COM
;           files.  Use deny-write sharing mode if read-only access to .BAT
;           files is requested.
;       3 = Use deny-none sharing mode on all compatibility-mode opens.
;           (default)
; 12  allow DOS LAN Services workstations to use a second NETBIOS session when
;     sending printer requests (default is 1)
; 13  number of 64KB buffers used for read-ahead
;       0-9 = number of buffers (default is 1)
; 14  convert incoming path specifications into most basic format (default is
0)
; 15  Oplock Timeout and NETBIOS Acknowledgment Timeout (default is 0)
;     0 = 35 second Oplock Timeout; 34 second NETBIOS Acknowledgment Timeout
;     1 = 70 second Oplock Timeout; 69 second NETBIOS Acknowledgment Timeout
;     2 = 140 second Oplock Timeout; 127 second NETBIOS Acknowledgment Timeout
;     3 = 210 second Oplock Timeout; 127 second NETBIOS Acknowledgment Timeout
;     4 = 280 second Oplock Timeout; 127 second NETBIOS Acknowledgment Timeout
;     5 = 350 second Oplock Timeout; 127 second NETBIOS Acknowledgment Timeout
;     6 = 420 second Oplock Timeout; 127 second NETBIOS Acknowledgment Timeout
;     7 = 490 second Oplock Timeout; 127 second NETBIOS Acknowledgment Timeout
;     8 = 560 second Oplock Timeout; 127 second NETBIOS Acknowledgment Timeout
;     9 = 640 second Oplock Timeout; no NETBIOS Acknowledgment Timeout
; 16  validate IOCTLs (default is 1)
; 17  how long the server maintains unused, dynamic big buffers before freeing
;     the memory.
;     0 = 0 seconds                  4 = 5 minutes       8 = 1 hour
;     1 = 1 second                   5 = 10 minutes      9 = indefinitely
;     2 = 10 seconds                 6 = 20 minutes
;       3 = 1 minute (default)         7 = 40 minutes
; 18  how long the server waits after failing to allocate a big buffer before
;     trying again.
;       0 = 0 seconds                  3 = 1 minute (default)
;       1 = 1 second                   4 = 5 minutes
;       2 = 10 seconds                 5 = 10 minutes
; 19  RAW read and RAW write SMB protocols (default is 1)
; 20  server responds to announcement requests (default is 1)
;
; The next lines help you to locate bits in the srvheuristics entry.
;                           1         2
;                  012345678901234567890
 srvheuristics = 111101411113110013311
 SRVSERVICES =
 srvnets = NET1
```

9.14.3 Die Replicator-Sektion

Innerhalb eines Netzwerkes wird vom IBM LAN Server Requester die automatische Dateireplikation durch den sogenannten Replicator-Service als Dienst zur Verfügung gestellt. Hierdurch kann ein automatischer Dateiabgleich für bestimmte Dateien bzw. Verzeichnisse durchgeführt werden. Man unterscheidet dabei zwischen einem Computer, der als Exporter agiert und die Daten zur Verfügung stellt, und einem anderen Computer, der die Importer-Funktion übernimmt und die Daten empfängt. Eine reine Requester-Maschine ohne gestartete Peer-Dienste kann nur die Funktion des Importers übernehmen, wohingegen derselbe Computer mit gestarteten Peer-Diensten sowohl Importer als auch Exporter sein kann.

Welche Funktion der einzelne Rechner innerhalb des Netzwerks ausfüllt, wird durch das Schlüsselwort *replicate=* in der Replicator-Sektion festgelegt. Hierbei darf bei einem reinen Requester nur der Wert IMPORT eingetragen werden.

Ein Requester mit gestarteten Peer-Diensten kann sowohl IMPORT, EXPORT oder auch BOTH als gültigen Wert besitzen. Wird der Wert BOTH eingetragen, kann der Computer sowohl als Importer als auch als Exporter fungieren. Hierbei ist jedoch zu beachten, daß die Pfadangaben für den Importpfad und Exportpfad unterschiedlich sind. Damit eine Dateiabgleich jedoch überhaupt stattfinden kann, muß der Replicator-Dienst gestartet sein. Dies kann über die Befehlszeile mit »Net Start Replicator« durchgeführt werden. Um den Replicator-Service automatisch mit dem Requester zu starten, muß diesem als zusätzlicher Service in der Requester-Sektion unter dem Schlüsselwort *wrkservices=* eingetragen werden (sofern der Computer nur als Importer agiert) beziehungsweise in der Peer-Sektion unter *srvservices=*, wenn auch die Exporter-Funktionalität genutzt wird.

Der Exporter-Computer

Um einen Computer als Exporter zu konfigurieren, ist für das Schlüsselwort *replicate=* entweder der Wert EXPORT oder BOTH anzugeben. Dies bedingt jedoch, daß die Peer-Dienste gestartet wurden. Ein Exporter-Rechner informiert bei Änderung der Dateien, die sich in dem unter *exportpath=* angegebenen Verzeichnis inklusive Unterverzeichnisse befinden, automatisch alle Rechner, die in der aktuellen Domäne als Importer konfiguriert worden sind. Soll die Replikation ganz gezielt mit bestimmten Rechnern oder Importer einer anderen Domäne durchgeführt werden, so ist das Schlüsselwort *exportlist=* in die Replicator-Sektion einzufügen. Hier können dann eine beliebige Anzahl von NetBios-Namen von Rechnern in der aktuellen Domäne oder andere Domänen angegeben werden. Existiert dieses Schlüsselwort in der »ibmlan.ini«, werden die Nachrichten, daß Dateien geändert wurden und eine Aktualisierung auf den Importern ansteht, an alle angegebenen Rechner und Domänen verschickt. Hierbei sind die Rechner- und Domänennamen durch ein »;« zu trennen. Es dürfen keine Leerzeichen verwendet werden. Soll zum Beispiel die Replikation mit den Rechnern REQ001, REQ005 und ADS001 in der aktuellen Domäne sowie mit allen Importern der Domäne TESTDOM1 durchgeführt werden, müßte der entsprechende Eintrag wie folgt lauten:

```
exportlist=REQ001;REQ005;ADS001;TESTDOM1
```

Über den Eintrag *exportpath=* wird festgelegt, welche Dateien für den Dateiabgleich verwendet werden sollen. Alle abzugleichenden Dateien müssen sich unterhalb dieser Pfadangabe befinden. Der Vorgabepfad hierfür ist »lw:\ibmlan\repl\export«. Wird dieses Schlüsselwort nicht angegeben, wird dieser Vorgabepfad verwendet. Zusätzlich läßt sich über eine Datei »repl.ini« (siehe unten) festlegen, wie der Abgleich der Dateien erfolgen soll.

Um den als Importer konfigurierten Computern mitzuteilen, wann ein erneuter Dateiabgleich erfolgen soll, kann der Exporter über diverse Parameter in der »ibmlan.ini« konfiguriert werden.

Als erster Parameter ist das Schlüsselwort *guardtime=* zu nennen. Hiermit wird festgelegt, wie lange sich die Dateien im *exportpath=* nicht verändert haben dürfen, bevor eine Benachrichtigung an die Importer gesandt wird. Wird beispielsweise ein Wert von 5 angegeben, dürfen sich die Dateien im Export-Pfad fünf Minuten lang nicht verändern, damit die Importer-Rechner benachrichtigt werden. Dieser Wert ist jedoch nur dann relevant, wenn in der »repl.ini« (siehe unten) der Wert für *integrity=* auf »Tree« gesetzt wurde. Ferner ist das erlaubte Maximum für diesen Wert die Hälfte der unter *interval=* angegebenen Zeitspanne. Der Wert von *interval=* wiederum legt fest, in welchem Zeitabstand in Minuten die Dateien und Unterverzeichnisse auf Änderungen hin überprüft werden sollen. Das erlaubte Maximum hierbei ist 60 Minuten.

Konnte ein Importer aus irgendeinem Grund die gesendete Information des Exporters, daß Dateien zum Abgleich anstehen, nicht empfangen, bedeutet das noch lange keine Inkonsistenz der Daten. Über das Schlüsselwort *pulse=* kann festgelegt werden, in welchem Minutentakt die Information erneut gesandt werden soll. Damit erhalten auch Importer die Möglichkeit, Ihre Daten zu aktualisieren, wenn sie die originäre Information nach Ablauf des letzten *interval*-Zeitraums verpaßt haben.

Um die Belastung eines Exporters beim Dateiabgleich zu reduzieren, kann als letzte Option über den Wert für das Schlüsselwort *random=* angegeben werden, nach wieviel Sekunden ein Importer einen Verbindungsaufbau durchführen soll. Das bedeutet, daß jeder Importer, der eine Verbindung zum Exporter aufnehmen will, vom Exporter den in *random=* angegebenen Wert übermittelt bekommt. Der Importer wählt nun eine beliebige Zeitspanne zwischen 0 Sekunden und dem übermittelten Wert, bevor er die Verbindung aufbaut und mit dem Dateiabgleich beginnt.

Damit ist die Konfiguration eines Exporters abgeschlossen.

Die »Repl.ini«

Die »Repl.ini« wird auf dem Exporter verwendet, um festzulegen, ob gesamte Unterverzeichnisse oder nur Dateien abgeglichen werden sollen. Dabei finden lediglich die zwei Schlüsselwörter *intergity=* und *extent=* Verwendung. Beide können nur den Wert »Tree« oder »File« annehmen. Daraus ergeben sich die folgenden vier Kombinationsmöglichkeiten:

```
extent = tree
integrity = file
```

Mit dieser Kombination werden – ausgehend vom Export-Pfad – alle Unterverzeichnisse auf Basis von Dateien repliziert. Das bedeutet, sind einzelne Dateien in einem beliebigen Unterverzeichnis über den in *guardtime=* angegebenen Zeitraum unverändert geblieben, werden die Importer-Rechner darüber informiert und ein Dateiabgleich auf Dateibasis kann erfolgen.

```
extent = tree
integrity = tree
```

Bei dieser Kombination wird – ausgehend vom Export-Pfad – auf Basis von Verzeichnissen repliziert. Das bedeutet, sind einzelne Unterverzeichnis über den in *guardtime=* angegebenen Zeitraum unverändert geblieben, werden die Importer-Rechner darüber informiert, und ein Dateiabgleich auf Verzeichnisbasis kann erfolgen.

```
extent = file
integrity = file
```

Wird für den Parameter *extent=* der Wert »file« angegeben, werden nur Dateien des in *exportpath=* angegebenen Verzeichnisses abgeglichen. Unterverzeichnisse werden ignoriert. Die obige Kombination bewirkt, daß auf Dateibasis ein Dateiabgleich des Export-Verzeichnis durchgeführt wird.

```
extent = file
integrity = tree
```

Soll das Export-Verzeichnis nur dann repliziert werden, wenn das gesamte Verzeichnis über den in *guardtime=* angegebene Zeitraum unverändert geblieben ist, ist die letzte Kombinationsmöglichkeit zu verwenden.

Wie aus obigen Beispielen zu entnehmen ist, wird über den Wert des Schlüsselworts *extent=* festgelegt, ob ein Dateiabgleich nur die Dateien (»file«) des Export-Verzeichnises berücksichtigen oder ob alle Unterverzeichnisse (»tree«) in den Dateiabgleich mit eingeschlossen werden sollen. Die Vorgabe ist »tree«.

Über den Wert von *integrity=* wird dagegen festgelegt, ob die Importer informiert werden sollen, wenn einzelne Dateien verändert wurden oder wenn ein Verzeichnis im in *guardtime=* angegebenen Zeitraum unverändert geblieben ist. Wird der Wert »file« (die Vorgabe) verwendet, wird *guardtime=* ignoriert.

Existiert keine Datei »repl.ini« im Export-Verzeichnis, findet kein Dateiabgleich statt.

Der Importer-Computer

Der Importer stellt das Gegenstück zu einem Exporter dar und ist zugleich der Computer, der sich die veränderten Dateien von einem Exporter-Rechner holt. Der Exporter stellt die Dateien und die Benachrichtigung lediglich zur Verfügung. Der Importer jedoch muß aktiv werden, um eine Kopie der Dateien auf dem Rechner zu erhalten. Aus diesem Grund gibt es für die Konfiguration des Importers einige zusätzliche Parameter.

Um als Importer agieren zu können, muß der Wert für *replicate=* entweder auf IMPORT oder auf BOTH gesetzt werden, wobei BOTH nur dann einen gültigen Eintrag darstellt, wenn die Peer-Dienste gestartet wurden.

Vergleichbar der Exportliste, die auf einem Exporter über das Schlüsselwort *exportlist=* festlegt, welche Rechner den überhaupt von einer Änderung benachrichtigt werden sollen, wird für den Importer über das Schlüsselwort *importlist=* festgelegt, von welchen Rechner bzw. von welchen Exportern in welcher Domäne überhaupt Dateien abgeholt werden sollen. Wird kein Wert angegeben, werden alle Benachrichtigungen von allen Exportern in der aktuellen Domäne berücksichtigt. Ansonsten finden lediglich die Benachrichtigen von den angegebenen Rechnern beziehungsweise von den Rechnern der angegebenen Domäne(n) Beachtung.

Die abzugleichenden Dateien werden in dem unter *importpath=* angegebenen Verzeichnis gespeichert. Wird kein Verzeichnis angegeben, wird das Vorgabeverzeichnis »lw:\ibmlan\repl\import« verwendet.

Damit Dateien von einem anderen Rechner abgeglichen werden können, ist eine entsprechende Berechtigung notwendig. Das bedeutet, daß der Importer sich an der Domäne oder an dem Peer-Server anmelden muß, um Zugriff auf die Dateien zu erhalten. Hierzu werden die Rechte »Lesen« und »Attribute« benötigt. Um nun Zugriff auf das Exportverzeichnis zu erhalten, gibt es mehrere denkbare Szenarien.

Ist kein Benutzer angemeldet, wird überprüft, ob das Schlüsselwort *logon=* in der Replicator-Sektion der »ibmlan.ini« gesetzt wurde. Ist dies der Fall, wird ferner überprüft, ob ebenfalls eine Angabe für das Schlüsselwort *password=* vorliegt. Beide Informationen werden dann verwendet, um eine Verbindung zum Exporter aufzubauen. Wurde das Schlüsselwort *logon=* nicht in die Replactor-Sektion eingetragen (die Vorgabe) und ist kein Benutzer angemeldet, wird versucht, eine Verbindung unter Verwendung des NetBios-Namens des Rechners aufzubauen. Dies bedingt also, daß entweder der NetBios-Name des Rechners oder die unter dem Schlüsselwort *logon=* angegebene ID einem gültigen Account auf dem Exporter entspricht. Da bei einem Peer-Netzwerk jeder Rechner seine eigene Benutzerprofilverwaltung besitzt, müßte ein Importer also auf allen als Exporter fungierenden Peer-Rechnern definiert sein.

Ist ein Benutzer angemeldet, hängt es vom Wert für das Schlüsselwort *tryuser=* ab, ob ein Dateiabgleich erfolgt oder nicht. Ist der Wert »yes«, wird versucht, unter Verwendung der aktuellen Benutzerkennung und des Benutzerpasswortes eine Verbindung zum Exporter aufzubauen. Dies ist dann erfolgreich, wenn der aktuelle Benutzer im Exportpfad die benötigten Rechte besitzt. Konnte keine Verbindung aufgebaut werden, findet so lange kein Dateiabgleich statt, bis der Benutzer sich abgemeldet hat und eine Verbindung entweder unter Verwendung des NetBios-Namens des Rechners oder der unter *logon=* angegebenen ID aufgebaut werden konnte.

Wurde der Wert auf »no« gesetzt, wird der Versuch des Verbindungsaufbaus zum Exporter nur durchgeführt, wenn kein Benutzer angemeldet ist.

Hier ein Beispiel für eine Importer-Konfiguration eines Requesters. Eine eventuell vorliegende Benutzerkennung soll nicht verwendet werden (*tryuser=no*) und die ID zum Verbindungsaufbau lautet »repluser« ohne Paßwort.

```
[replicator]
replicate = IMPORT
importpath = C:\ibmlan\repl\import
```

```
tryuser = no
logon = repluser
password =
```

Die Services-Sektion

Die Services-Sektion übernimmt die Zuordnung der ausführbaren Programme zu den in den anderen Sektionen verwendeten Namen der Dienste. Hierbei kann ein absoluter Pfad inklusive der ausführbaren Datei angegeben werden oder ein relativer Pfad ausgehend von dem Verzeichnis »\ibmlan«. Hier sollten keine Änderungen vorgenommen werden, um zu gewährleisten, daß die angegebenen Dienste auch ordnungsgemäß ausgeführt werden können.

```
[services]
 messenger = services\msrvinit.exe
 peer = services\peerinit.exe
 replicator = services\replicat.exe
 requester = services\wksta.exe
```

10 Systems Management

von Oliver Mark

In einer verteilten EDV-Umgebung ist es ab einer gewissen Anzahl von Geräten (egal ob Server, Clients oder Peripherie) unabdingbar, diese in einer gewissen Form zu steuern. Wie weit dieses Steuern geht, hängt zum einen von der Leistungsfähigkeit der Maschinen und des Netzes ab, zum anderen auch von der Notwendigkeit und Relevanz. Das heißt, ein reiner Testarbeitsplatz muß nicht in gleicher Form überwacht werden wie zum Beispiel ein Abteilungssserver mit kritischen Daten und Anwendungen. Im folgenden wird auf System-Management allgemein, sowie auf den System-Management-Client und dessen Funktionen eingegangen.

Zuerst stellt sich die Frage: Was versteht man unter Systems Management ?

– Die Überwachung von Geräten

Je nach Aufgabe der einzelnen Geräte einer Infrastruktur, müssen diese auch unterschiedlich bei der Überwachung gewichtet werden. So sind die Festplatten eines zentralen Servers mit Sicherheit von größerer Wichtigkeit als ein externes CD-ROM-Laufwerk an einem Arbeitsplatz. Welche Komponenten bei Geräten konkret überwacht werden können, hängt von Bauart und Hersteller ab, dies sind unter anderem: Hauptspeicherbausteine, Festplatten (SCSI, RAID), Adapterkarten, angeschlossene Peripheriegeräte.

– Die Überwachung von Anwendungen und Daten

Für einige Anwendungen in einer vernetzten Umgebung ist es extrem wichtig, daß diese permanent zur Verfügung stehen bzw. daß die bereitgestellten Daten korrekt sind. Deshalb ist es notwendig, bestimmte Prozesse auf ihren Status zu überprüfen (ist Prozess XY auf dem Server YX im korrekten Status) und Datenbestände (Files, Datenbanken) auf ihr Vorhandensein (existiert die Datei Config.Sys) und auf ihren Zustand (wer hat wann was daran geändert) zu prüfen.

– Die Überwachung und Steuerung von Performance und Volumina

Hierunter fällt zum einen die Planung eines gesamten Netzes hinsichtlich Benutzerzahl und Rechnerkapazität, sowie die permanente Kontrolle dieser Daten. So sind hier sicherlich Prozessorauslastungen, Festplattenplatz, Netzwerkbelastung wichtige Kriterien der Überprüfung.

– Die zentrale Pflege von Software und Anwendungen

Bei steigender Größe eines Netzwerkes steigt (man munkelt: exponential) die Komplexität der installierten Software und der bereitgestellten Daten. So ist es zwangsläufig sinnvoll, Standardisierungen vorzunehmen. Dies kann zum einen auf freiwilligen Basis mittels Empfehlungsschreiben an die Benutzer geschehen, oder aber durch Einsatz von geeigneten Tools. Einige davon fallen in die oben genannten Kategorien der Überwachung, andere greifen aktiv in ein intalliertes Netzwerk ein. So auch die Verteilung von Daten, Applikationen und Betriebssystemen von einer zentralen Stelle aus, nämlich der Softwareverteilung.

– Die Unterstützung der Benutzer

Trotz aller Automatisierung bleibt es nicht aus, daß ein Benutzer eines Gerätes Probleme mit dem Umgang mit ihm hat. Dann muß sinnvollerweise an einer zentralen Stelle Know-how bereitgestellt werden, um ihm zu helfen. Dies kann als Wissensdatenbank oder in persona (User Help Desk) geschehen.

– Die Sicherung von Daten, aber auch Geräten

Hierunter fällt die mittlerweile fast triviale Datensicherung (zum Beispiel Bandlaufwerk), sowie der Schutz von Geräten vor Diebstahl.

Systems Management bei der IBM

Dies soll keine vollständige Aufzählung aller IBM-Produkte auf allen vorhandenen Betriebssystemplattformen werden, sondern einen kurzen Blick auf die Historie in der PC-Welt darstellen. Mit Verfügbarkeit von OS/2 Warp Version 3 wurde im Bonuspak die erste Version von Systems Management unter OS/2 präsentiert: Das System Information Tool. Einige Funktionen, sowie das Look-and-feel auf der Arbeitsoberfläche stammen bis heute davon ab.

Einige Zeit später wurde aus dem System Information Tool ein eigenes Produkt mit dem Namen »Netfinity«, das bereits hier die Version 3.0 trug, und dann auch bis zu einer Version 3.06 weiterentwickelt wurde. Zeitgleich wurde das Produkt in »Systemview für OS/2« und kurzfristig auch in »PC-Systemview« umbenannt. So wurde es zum Beispiel im Warp Server 4 als Systemview 1.0 ausgeliefert.

Im Rahmen des Zukaufs der Firma Tivoli durch die IBM wurden sämtliche Produkte neu benannt, und das Systems Management unter OS/2 erhielt den Namen »TME10 Netfinity Manager für OS/2 4.03«. Im August 1996 schließlich wurde im Rahmen der IBM-Software-Server der TME10 Netfinity Server für OS/2 4.0 angekündigt. Gleichfalls durch den Zukauf von Tivoli wurde die Programmierung der Anwendungen auf den Tivoli-Framework umgestellt, der rein objektorientiert aufgebaut ist, und der IBM einiges an eigener Entwicklungsarbeit in diesem Gebiet erspart hat.

Die traditionellen IBM-Systems-Management-Disziplinen wurden gleichfalls in diesem Zuge über den Haufen geworfen, sodaß aus den »alten« sechs Disziplinen vier »neue« wurden: Availability, Operations & Administration, Security und Deployment. Zu den einzelnen Disziplinen gehören:

Deployment

– Softwareverteilung

– Inventory Management (Hardware-Inventarisierung)

– Asset Management (Kostenverteilung auf Benutzer)

Availability

– Event Automation (automatische Verfahren aufgrund bestimmter Zustände oder Meldungen)

– Mid-Level-Manager (Weitergabe von Informationen eines Netzes an eine übergeordnete Stelle (zum Beispiel Workstations meldet an Server, Server meldet an Großrechner)

– Network Management (Überwachung von Netzwerkkomponenten, wie zum Beispiel Hub, Router, Bridges)

– Performance-Management

– Performance Analysis/Reporting

Security

– User-Administration

– Audit-Analyse

Operations & Administrations

– Job Scheduling (Abarbeitung einer Prozedur zu einem bestimmten Zeitpunkt)

– Remote Control (Die passive oder aktive Steuerung eines entfernten Gerätes)

– Help Desk (Benutzerunterstützung im Problemfall)

– Backup & Restore (Datensicherung und Datenwiederherstellung)

– Output-Management (Druckersteuerung)

Diese vier Disziplinen finden sich natürlich im Systems-Management-Produkt TME10 Netfinity wieder, oder umgekehrt: Die Funktionen von TME10 Netfinity decken alle Bereiche des Systems Management nahezu vollständig ab.

10.1 TME10 Netfinity

Die Installation wird bei der Gesamtinstallation der Netzwerkkomponenten durchgeführt.

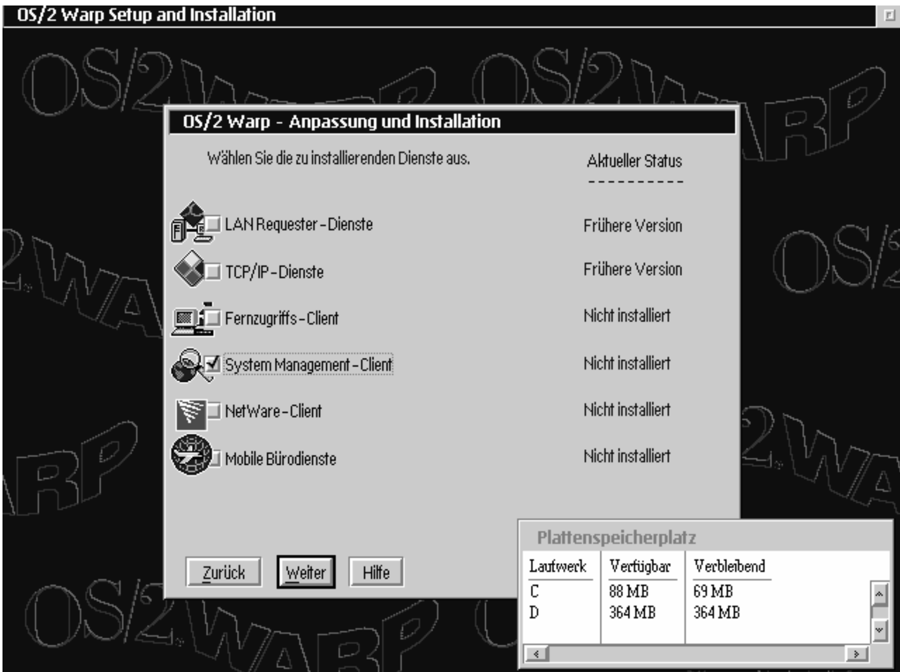


Abb. 10.1: Netzwerkinstallation – Systems-Management-Client

Im Auswahlbild der Netzwerkomponenten wählen Sie die Checkbox vor dem Eintrag *Systems Management-Client* aus. Sollten Sie eine nachträgliche Installation vornehmen wollen, so kommen Sie gleichfalls auf dieses Auswahlbild über die Ordner *System| Systemkonfiguration|Installieren/Entfernen|Netzwerkinstallation Anpassen*.

Während der Installation der Netzwerkomponenten werden Sie nach der Betriebsart Ihres Systems-Management-Clients gefragt.



Abb. 10.2: Betriebsart

Diese drei Auswahlmöglichkeiten sind im folgenden entscheidend darüber, ob sie diesen Client als eigenständiges Gerät betreiben wollen, das heißt, ohne Netzwerkanbindung, oder als Client in einem größeren Netzwerk. Die erste Auswahl installiert eine lokale Version, mit der sie keine Netzwerkverbindung und -überwachung betreiben können, jedoch die meisten der Funktionen lokal ausführen können. Die Netzwerkvariante unterscheidet sich in passiven Client und aktiven Client. Bei der passiven Variante wird der Client, ohne Ordner und Symbole auf der Arbeitsoberfläche anzulegen, installiert. Der lokale Benutzer hat also keine Möglichkeit, seinen PC zu managen. Die aktive Variante installiert zusätzlich zu der Netzwerkanbindung des Clients auch die lokale Verwaltung und legt den entsprechenden Ordner auf der Arbeitsoberfläche an.

Mit der Auswahl von *Installieren* beginnt die eigentliche Installation. Gegen Ende der Installation erhalten Sie ein weiteres Fenster mit möglichen Einstellungen.

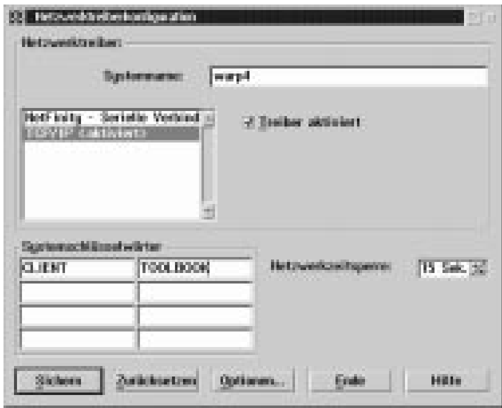


Abb. 10.3: Konfiguration

Im oberen Bereich können Sie den Systemnamen eingeben. Dieser ist üblicherweise eine sprechende Bezeichung des PC. Bei einer Netzwerkinstallation sollte hier der Systemname mit dem Hostnamen (bei TCP/IP) oder dem Maschinennamen (bei NETBIOS) übereinstimmen.

In mittleren Feld erhalten Sie eine Auswahl zu den Protokollen, die Sie aktivieren können. Hier erscheinen lediglich Protokolle, die auch auf dem Rechner verfügbar sind, das heißt, diese Box ist abhängig von den im Vorfeld installierten Netzwerk-Komponenten. Grundsätzlich werden TCP/IP, Netbios (inkl. Netbios über TCP/IP), IPX und serielle Schnittstelle unterstützt. Als Administrator eines Netzwerkes sollten Sie sich überlegen, welches Protokoll Sie für Ihr Systems Management auswählen; dies hängt sicherlich von der Umgebung und den bereits installierten Protokollen ab. Nur sollten Sie sich für ein Protokoll entscheiden. Als Benutzer werden Sie die Vorgaben Ihres Netzwerkes übernehmen müssen.

Im unteren Teil nun können Sie Ihr System mittels acht Systemschlüsselwörtern (Keywords) detailliert beschreiben. Diese Schlüsselwörter dienen zu einem späteren Zeitpunkt zur Identifikation und Gruppierung von Client von einer zentralen Konsole aus. Welche Sie hier eingeben können und dürfen, hängt von Ihrem Netzwerkadministrator ab. Sie sollten jedoch beachten, daß hier eine Unterscheidung von Groß- und Kleinbuchstaben gemacht wird (Client <> CLIENT).

Unter dem Schalter *Optionen* verbirgt sich die Möglichkeit, einen Netzwerkclient zu sichern.

Wählen Sie den oberen Checkbutton aus, um eine ferne Anmeldung zu erzwingen. Das bedeutet, daß ein zentraler Administrator, um auf Ihren PC zu kommen, eine Userid und ein Password eingeben muß. Der zweite Checkbutton dient zur Aktivierung eines Dienstes, der dem Benutzer eines Clients anzeigt, daß eine ferne Anmeldung vorliegt.



Abb. 10.4: Installation – Optionen

Haben Sie die Parameter der Netzwerktreiber geändert, erhalten Sie eine Information, daß diese Änderungen erst mit dem Neustart von Netfbase.Exe aktiv werden. Die Datei Netfbase.Exe ist der zentrale Client-Prozeß, der grundsätzlich aktiv sein muß, um TME10 Netfinity benutzen zu können.

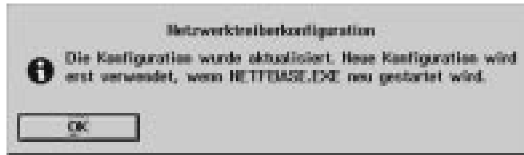


Abb. 10.5: Netzwerktreiber

Beenden Sie das Installations-Fenster, indem Sie zunächst *Sichern* und danach *Ende* anklicken. Nun erscheint eine Meldung, daß Ihre Config.Sys aktualisiert werden muß, und was dort geändert werden soll.

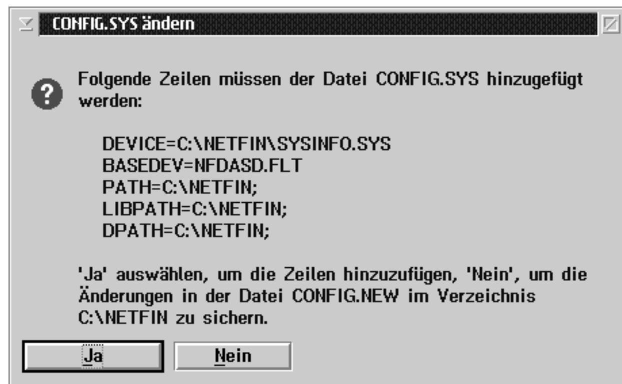


Abb. 10.6: Config.Sys ändern

Sie können hier ruhigen Gewissens *JA* angeben. Darauf folgt eine weitere Meldung, daß die Änderungen durchgeführt wurden, und daß Sie nun einen Neustart durchführen müssen. Außerdem weist Sie die Installation darauf hin, daß Sie nach dem Neustart den Security Manager aufrufen sollen. Dies hat den Hintergrund, daß die Grundeinstellung aller Sicherheitsregeln im TME10 Netfinity *disabled* sind. Konkret bedeutet dies, daß jeder, der Zugriff auf eine TME10-Netfinity-Manager Konsole hat, *alles* mit den zugänglichen Clients im Netz machen kann; und was das alles ist, werden Sie im folgenden noch sehen.

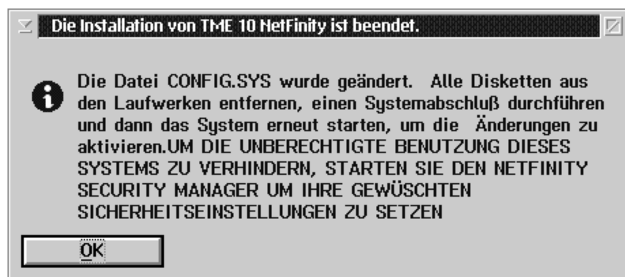


Abb. 10.7: Neustart

10.1.1 Planung

Bei der Planung sollten Sie einige wichtige Punkte nicht vergessen:

– Eine vollständige Installation benötigt ca 6,5 Mbyte Plattenplatz.

– Der Systems-Management-Client belegt ca 0,5 bis 1 Mbyte Hauptspeicher im laufenden Betrieb.

– Ihr Netzwerk sollte bei einer entsprechenden Netzwerkinstallation genügend Kapazitätsreserven haben, denn alle Überwachungsprozesse melden sich in definierten Intervallen bei ihrer konfigurierten Zentrale.

– Alle Sicherheitsmechanismen sind bei der Installation inaktiv.

10.1.2 Installationsmethoden

Die Installation des Systems-Management-Clients kann auf mehrere Arten geschehen:

– Die Installation erfolgt im Rahmen einer Neuinstallation oder einer selektiven Nachinstallation über den oben beschriebenen Weg mittels CD-ROM oder Diskette.

– Die Installation erfolgt durch den Benutzer, allerdings als nachträgliche Netzwerkinstallation von einem zentralen Installationsserver. Hierbei ist es möglich, den Benutzer in die Installation aktiv eingreifen zu lassen (attended installation) und ihn alle Angaben, wie oben beschrieben, selbst machen zu lassen.

– Oder dem Benutzer werden die Installationsparameter in einer Antwortdatei bereits vorzugeben, so daß er lediglich die Installation starten muß, und dann ohne Menüführung eine definierte Installation erhält (lightly attended installation).

– Die Installation wird von einer zentralen Stelle mittels eines Softwareverteilungsverfahren angestoßen. Hier hat der Benutzer weder die Möglichkeit, Parameter anzugeben, noch kann er entscheiden, wann die Installation durchgeführt wird (unattended installation).

Für die Installationsarten 2, 3 und 4 wird mit dem Programm eine Antwortdatei Netfbase.Rsp mitgeliefert, die alle Angaben der Installationsfenster als Parameter kennt. So zum Beispiel:

```
; Beispielantwortdatei für die ferne Installation von TME10
; Selection – Objekte, die im Hauptmenü ausgewählt werden.
; 0 – Eigenständiger Betrieb (Keine Netzwerkunterstützung)
; 1 – Passiver Client-Betrieb
; 2 – Aktiver Client-Betrieb
Selection = 0
; InstallTo – Dies ist das Verzeichnis auf dem Client-
; Ziellaufwerk, auf dem TME10 Netfinity installiert wird.
InstallTo = \NETFIN
```

Diese Antwortdatei wird während der Installation abgearbeitet. Für eine einheitliche Installation in einem Netzwerk empfiehlt sich, für alle Benutzer die gleiche Antwortdatei, oder eine weitgehend ähnliche zu verwenden (zum Beispiel Unterschiede in den Schlüsselwörtern), um

eine definierte Umgebung zu erhalten.

Eine solche Installation wird folgendermaßen aufgerufen:

```
Installationspfad\NETFINST
  /S:Quellaufwerk
  /TU:ZiellaufwerkClient
  /X   (Remote Ausführung)
  /A:I (Aktion Install, Löschen mit /A:D)
../L1:logfilelaufwerk\netfinst.lg1
  /L2:logfilelaufwerk\netfinst.lg2
  /R:quellaufwerk\NETFINST.RSP
```

Also bei einer Installation vom Serverlaufwerk X:\img\netfin auf den Client WARP4, Laufwerk C:

```
X:\IMG\NETFIN\NETFINST
  /S:X:\IMG\NETFIN
  /TU:C
  /X
  /A:I
  /L1:Y:\LOG\WARP4.LG1
  /L2:Y:\LOG\WARP4.LG2
  /R:\X:\IMG\RSP\WARP4.RSP
```

Nach einer erfolgreichen Installation und Neustart findet sich ein neues Icon im Systemstart-Ordner wieder. Leider wird bei der Installation das echte Objekt »TME10-Netfinity|Netzwerkschnittstelle« dort angelegt. Verschieben Sie deshalb das Original in den Ordner *TME10 Netfinity* auf der Arbeitsoberfläche und legen Sie im Systemstartordner eine Referenz dieses Objektes an.

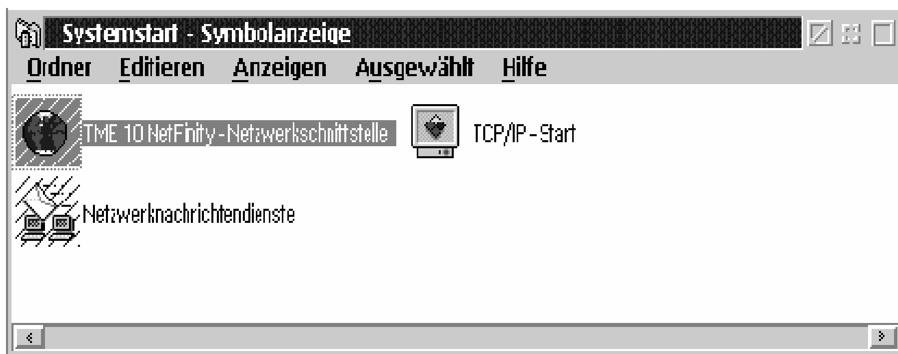


Abb. 10.8: Systemstart-Ordner

Im Hintergrund wird ein Fenster mit dem entsprechenden Prozeß gestartet. Hierbei handelt es sich um die bereits erwähnte Netfbase.Exe



Abb 10.9: Fenster *Netzwerkschnittstelle*

10.1.3 TME10-Basisfunktionen

Nach der Installation finden Sie, wie oben bereits erwähnt, ein neues Symbol im Systemstart-Ordner und ein aktives Hintergrundfenster mit der Netzwerkschnittstelle. Außerdem ein neues Symbol auf der Arbeitsoberfläche: Den Ordner *TME10 Netfinity*. In ihm befinden sich drei Symbole. Sollten Sie sich für den passiven Client-Betrieb während der Installation enschieden haben, finden Sie zwar das Symbol im Systemstartordner, jedoch keinen Ordner auf der Arbeitsoberfläche.



Abb. 10.10: TME10-Ordner

In umgekehrter Reihenfolge beginnend, rechts das Symbol für die Netzwerktreiberkofiguration, mit dem Sie wieder auf das bereits bekannte Bild der Netzwerktreibereinstellungen kommen.

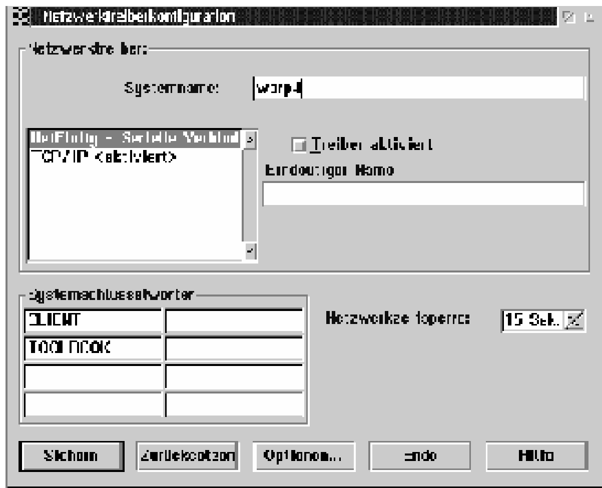


Abb. 10.11: Netzwerk-schnittstellenkonfiguration

Das mittlere Symbol ruft den Systemeditor mit der README-Datei von TME10 Netfinity auf. Dort finden Sie Hinweise auf Änderungen im Programm oder Probleme bei Installation oder Betrieb, sowie Hinweise auf Unverträglichkeiten mit anderen Programmen, die erst nach Redaktionsschluß der Handbücher bzw. Begleitliteratur bekannt wurden.

Das Symbol zur linken schließlich startet den lokalen Serviceteil des Programms, der alle verfügbaren Komponenten anzeigt. Beim Start erscheint das bereits von TME10 Netfinity bzw. Systemview bekannte Logo, um den TME10-Schriftzug erweitert.

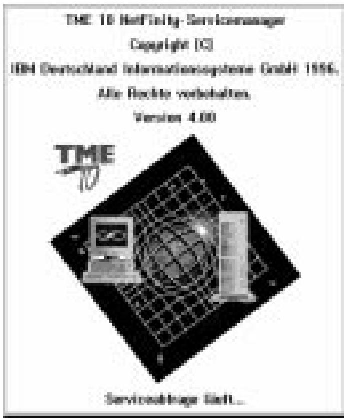


Abb. 10.12: TME10-Netfinity-Logo

Diese können sich in Abhängigkeit des benutzten PCs unterscheiden. So wird beispielsweise ECC-Memory-Setup nur dann angezeigt, wenn dieser physikalisch vorhanden ist, seien Sie also nicht verwundert, wenn Ihr Servicemanager-Ordner anders aussieht als hier gezeigt. Alle

bekannten Funktionen werden im folgenden beschrieben.

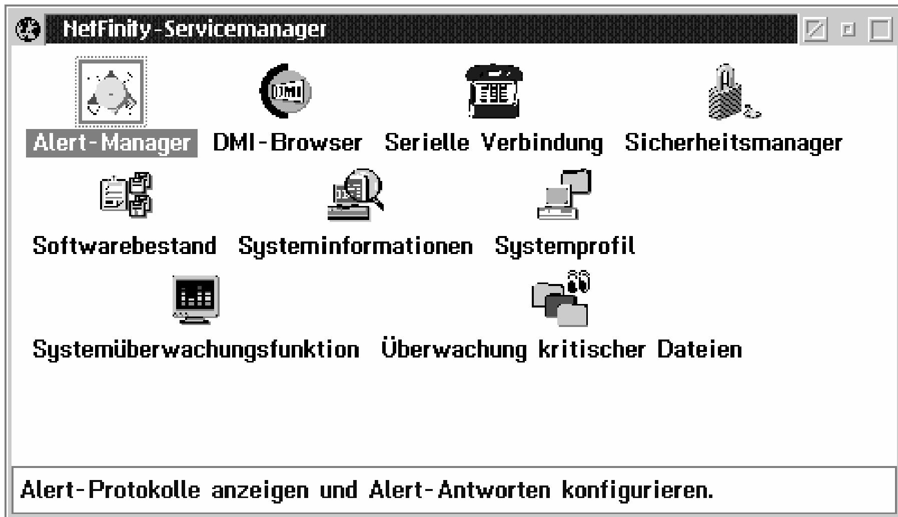


Abb. 10.13: Servicemanager

Hier nochmals zur Begriffsvertiefung:

Es gibt zum einen ein Software Produkt »TME10 Netfinity Manager für OS/2«, was prinzipiell die zentrale Konsole zum Systems-Management-Client darstellt. Andererseits heißt die Schaltfläche Ihres lokalen Clients gleichfalls »Manager«, nämlich »Service-Manager«. Dies bedeutet lediglich, daß Sie selbst in der Lage sind, bestimmte Netfinity-Services zu steuern. Solten Sie keine lokalen Ordner haben, das heißt, Sie haben einen passiven Client installiert, kann Ihr PC nur von einem entfernten TME10-Netfinity-Manager gerwaltet und überwacht werden.

Serial Control

Die Freigabe zur Überwachung eines PCs erfolgt für Netzwerkprotokolle über deren Konfiguration und damit der Aktivierung im Fenster *Netzwerktreiber*. Da aber eine Überwachung oder Steuerung eines PCs über eine serielle Verbindung meist nur eine temporäre Sache ist, wird dieser Zugang separat konfiguriert. Hierzu ist es natürlich notwendig, ein Modem am konfigurierten COM-Port angeschlossen zu haben.

Der vorgegebene Name einer solchen Konfiguration lautet »Automatische Antwort«, das heißt, der Client antwortet, sobald er von einer entfernten Stelle angerufen wird. Im Feld *Nummer* können Sie gegebenenfalls eine Nummer eingeben, die berechtigt ist, die Netfinity-Services dieses PCs zu benutzen. Geben Sie hier keine Nummer ein, sind grundsätzlich alle Anrufer berechtigt, die Services zu nutzen. COM-Port und COM-Baudrate setzen Sie entsprechend Ihrer Modemvorgaben. Momentan können Sie über diesen Weg keine ISDN-Zugänge konfi-

gurieren. *Benutzer-ID* und *Kennwort* verhalten sich analog zum Nummernfeld. Sie können hier einen bestimmten Benutzer mit einem Kennwort definieren, oder die Felder frei lassen, wenn alle Benutzer Zugang haben sollen.

Die Check-Box *Autom. Start nach Systemstart* bewirkt analog zu den Netzwerktreibern einen Eintrag im Systemstart-Ordner.

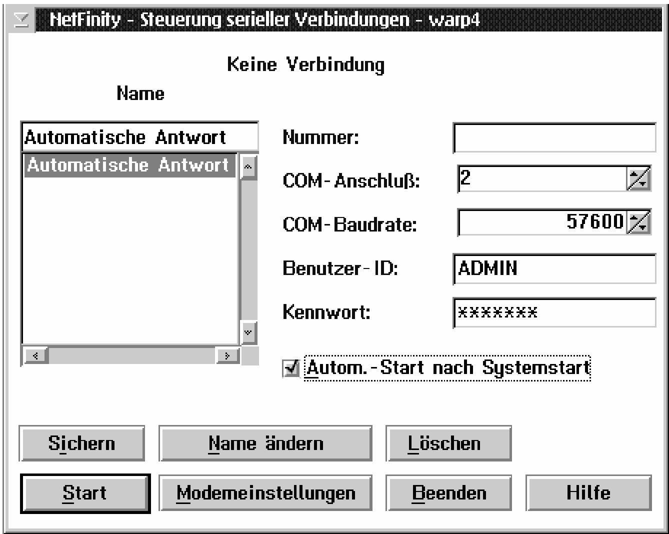


Abb. 10.14: Serielle Verbindung

Über den Schalter *Modemeinstellungen* können Sie Ihr Modem und Initialisierungs-, sowie Beendigungskommando einstellen (*Initstring* und *Hangupstring* sind hier die gängigen englischen Begriffe).

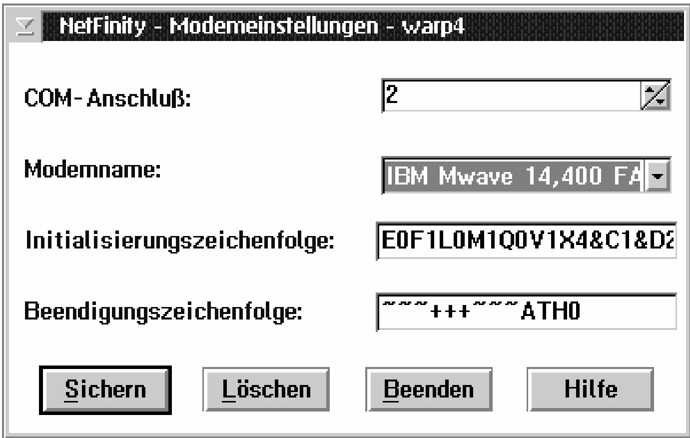


Abb. 10.15: Modemeinstellungen

Zurück im Fenster *Serielle Verbindungen* können Sie nun mit dem Schalter *Start* das Hintergrundprogramm sofort starten; Sie dürfen dann allerdings das Fenster nicht schließen, da sonst automatisch die Anrufüberwachung deaktiviert wird.

Security Manager/Sicherheitsmanager

An manchen Stellen ist sich hier die Übersetzung nicht ganz schlüssig, die Funktion nun mit Security- oder Sicherheitsmanager zu betiteln. Was die Funktion jedoch tut, ist recht einfach: Sie verwaltet die Benutzer, die von einem entfernten System Zugang zum lokalen System bekommen sollen. Dies ist unter anderem an der Titelleiste zu erkennen *Eingehende Kennwörter*. Im Gegensatz dazu können auf einer Manager-Version von TME10 Netfinity auch ausgehende Kennwörter angegeben werden, um so zum Beispiel einen Zugriff zu automatisieren.

Im Feld *Benutzer-ID* definieren Sie den Benutzer und weisen ihm über *Kennwort* und *Prüfkennwort* ein Paßwort zu. Im Feld *Services* definieren Sie, welche der lokalen Services vom jeweiligen Benutzer genutzt werden darf. Die Check-Box *Sicherheitsmanagerzugriff* gestattet dem Benutzer den Zugriff auf genau dieses Fenster und gibt ihm damit die Möglichkeit, selbst Benutzer anzulegen.

Standardmäßig ist ein Benutzer <BENUTZER> definiert, der kein Kennwort hat; diesen sollten Sie umgehend löschen. Legen Sie einen Admin-Benutzer an, der alle Funktionen im Zugriff hat, damit Sie selbst gegebenenfalls die Möglichkeit haben in Ihren PC einzugreifen. Alle weiteren Benutzer sind beliebig definierbar.

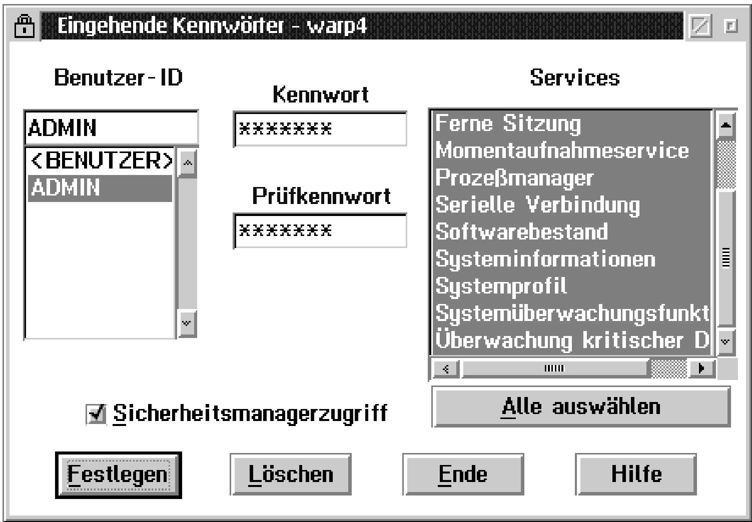


Abb. 10.16: Sicherheitsmanager

10.1.4 Deployment-Funktionen

Was sind »Assets«?

Wie schon im Eingangskapitel beschrieben, teilen sich die einzelnen Funktionen von TME10 Netfinity in verschiedene Schwerpunkte auf. Einer davon sind die sogenannten Deploymentfunktionen, die zur Verwaltung von Assets dienen, oder im Klartext: Dienste, die zur Sicherung von materiellem und geistigen Eigentum zählen. Darunter zählen zum Beispiel Hardwarekomponenten, Betriebssystemsoftware, Anwendungen, wichtige Daten und wichtige Prozesse (laufende Anwendungen).

System-Profile

Im System-Profil werden alle Informationen eingetragen, die zur genauen Beschreibung des PCs und dessen Benutzers dienen. Auf den einzelnen Seiten werden jeweils zu System, Benutzer, Standort, Kontakte und Verschiedenes Einträge gemacht.

System-Seite 1 bis 3

Hier werden Eintragungen zum Gerät gemacht. Einen Teil der Informationen erkennt der Service automatisch durch Hardwareerkennung (zum Beispiel Seriennummer). Der Großteil muß allerdings manuell erfaßt werden.

Auf der ersten Seite der System-Seiten wird das Gerät selbst erfaßt, auf Seite 2 Bildschirm und Drucker, und auf Seite 3 das Modem.

Abb. 10.17: System-Profile-Systemdaten Seite 1

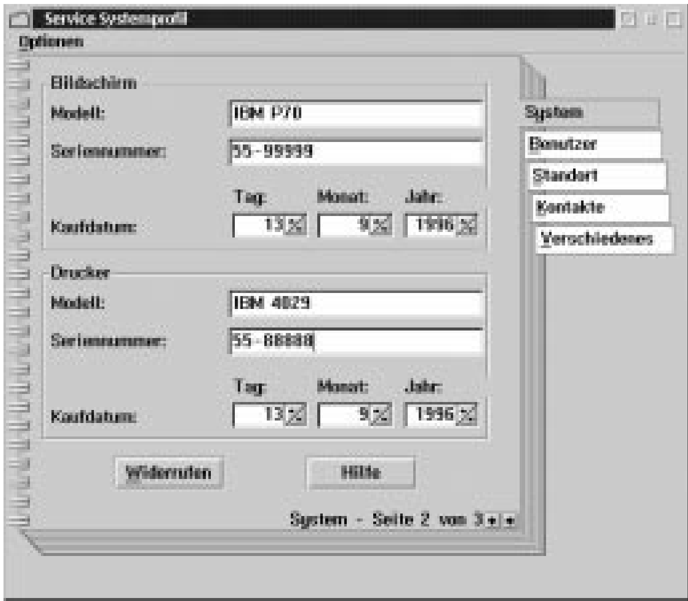


Abb. 10.18: System-Profile-Systemdaten Seite 2

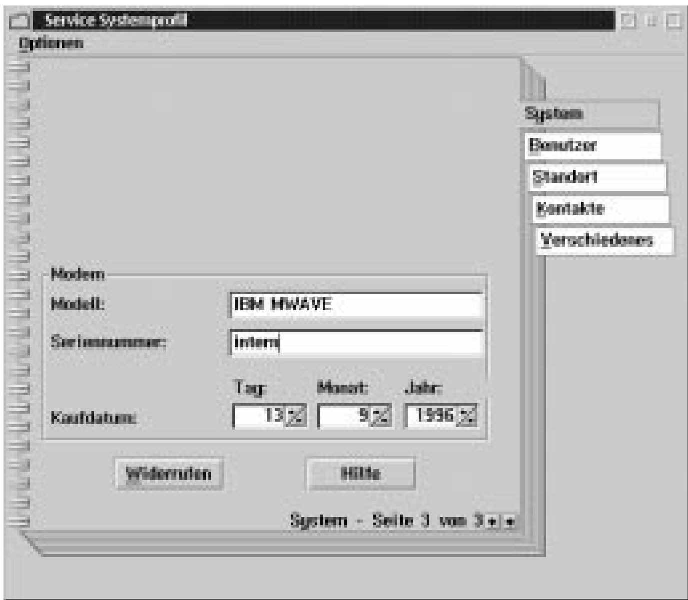


Abb. 10.19: System-Profile-Systemdaten Seite 3

Benutzer-Seite 1 bis 3

Hier werden genaue Angaben zum Benutzer gefordert: Vorname, Name, Mitarbeiter-ID, Titel, Abteilungsname, Abteilungsnummer, Bereich. Auf der zweiten Seite kann Schichtarbeit und geplante Arbeitszeiten agegeben werden. Auf der dritten Seite schließlich die Privatadresse.



Abb. 10.20: System-Profile Benutzerdaten Seite 1



Abb. 10.21: System-Profile Benutzerdaten Seite 2

Abb. 10.22: System-Profile Benutzerdaten Seite 3

Standort

Hier wird die Firma mit allen Adreßeinzelheiten (Straße, Stadt, Bundesland, Postleitzahl, Staat), sowie Angaben zum einzelnen Gerät (Standortname, Büronummer, Gebäude, Stockwerk) gemacht.

Kontakte, Seite 1 bis 2

Eintragungen, wie Sie den Benutzer über die verschiedensten Kommunikationswege (Telefon, Fax, Pager, Handy, Email) erreichen können, nehmen Sie auf der ersten Seite vor.

Auf der zweiten Seite haben Sie die Möglichkeit, Ihr gesamtes Umfeld freizulegen (Stellvertreter, Techniker, Manager und Sekretärin).

Verschiedenes, Seite 1 bis 3

Hier können Sie in Fließform alle Zeilen mit Gedichten, Liedern oder sonstigen zusätzlichen Informationen füllen.

Zu guter Letzt können Sie alle Informationen mittels des *Auswahl Optionen* entweder auf Ihrer lokalen Maschine aktualisieren (Netfinity intern), die eingetragenen Informationen in eine ASCII-Datei kopieren, oder aber den Service verlassen, ohne die Daten zu speichern, denn ein Schließen den Fensters bewirkt automatisch eine Aktualisierung der TME10-Netfinity-Daten.

System-Information

Der Service *System Information* ist der Ursprung von TME10 Netfinity und tauchte erstmals im Bonuspak von Warp Version 3 auf. Hier sind alle systemrelevanten Daten zusammengefaßt, die sich automatisch in irgendeiner Form erkennen lassen. Erkennbar ist dies durchaus noch im Startfenster.



Abb. 10.23: Systeminformationen-Logo

Im Hauptfenster werden systemabhängig die erkannten Informationen angezeigt und können von dort aus detailliert betrachtet werden. Über die Option *Datei* können die Informationen des jeweiligen Fensters (oder alle Informationen aus dem Hauptfenster) extern gespeichert werden. Je nach installierter Umgebung sind das: Datei, Drucker, DB2/2 oder Lotus Notes.

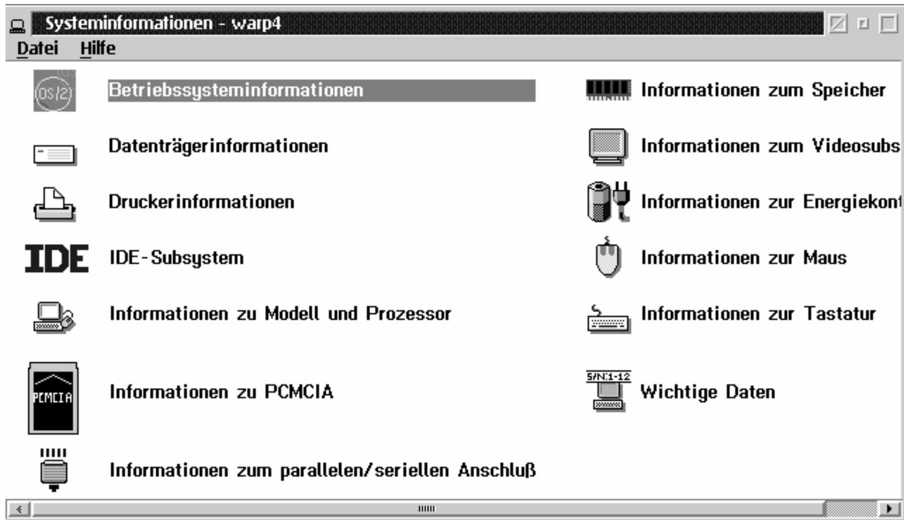


Abb. 10.24: Systeminformationen

Betriebssysteminformationen (Erkennung über Syslevel-Dateien)

– Zum Betriebssystem selbst: Version, Startlaufwerk, Zeitscheibe, Sitzung,...

– alle aktiven Programme in der Fensterliste

– CSD-Stufen der einzelnen Komponenten

– Die CONFIG.SYS-Informationen (detailliert mit Datum, Uhrzeit und Größe der zu ladenden Treiber.

– Datenträgerinformationen (Erkennung analog *qsystem*)

– Druckerinformationen (Erkennung anhand der installierten Druckertreiber)

– IDE/SCSI/RAID-Subsysteme (Erkennung anhand des Kontrollers)

– Modell & Prozessor (Erkennung analog *qsystem*)

– PCMCIA-Informationen (Erkennung über Card-Direktor-API)

– Informationen zum Speicher

– Informationen über Videosubsystem (Erkennung analog *qsystem*)

– Informationen zu Maus

– Informationen zur Tastatur

– Wichtige Daten (Seriennummer, Board-Revision,...)

Softwarebestand

Im Sinne einer Lizenzverwaltung (zumindest auf halbmanuellem Weg) ist es wichtig, die installierten Softwarekomponenten eines Systems zu kennen. Dies kann bei OS/2-Standardprodukten gegebenenfalls auch über die Systeminformationen herausgefunden werden. Sind allerdings Anwendungen ohne Syslevel-Kennung installiert, oder gar eigenentwickelte Software vorhanden, so muß dieser Softwarebestand gleichfalls erkennbar sein. Sehr viele Anwendungen sind bereits in der Migrationsdatenbank enthalten, die in die TME10-Netfinity-Datenbank übernommen wurden.

Einen gute Übersicht erhält man mittels der Option *Bestandsaufnahme|Vollständige Verzeichnissuche*. Hier werden mittels der Standard-Datenbank alle Verzeichnisse nach bekannten Programmen durchsucht.

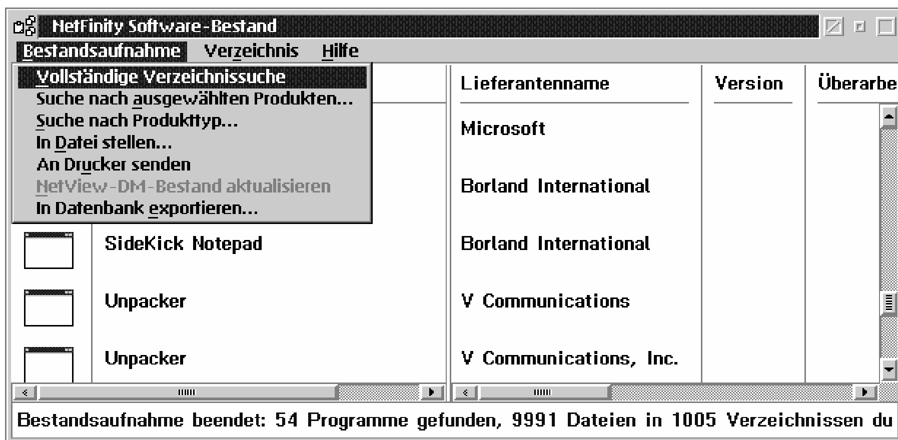


Abb. 10.25: Softwarebestand – Vollständige Suche

Über die Auswahl *Suche nach ausgewählten Produkten* kann die Suchliste auf ein oder mehrere Programme aus der Datenbank beschränkt werden.

Über die Auswahl *Produkttyp* kann nach verschiedenen Programmtypen (Netzwerk, DFÜ, Textverarbeitung, Grafik, Post, Server, Finanz,..) gesucht werden.

Auch hier kann ein Export der Daten in eine Datei oder in DB2/2 bzw. Lotus Notes erfolgen.

Über die Option *Verzeichnis* können neue Datenbanken angelegt, bestehende geändert oder gelöscht werden, sowie externe Datenbankformate eingelesen werden, wie zum Beispiel ein vorhandener NetviewDM/2-Katalog.

Will man eigene Produkte registieren, so ist sinnvoll den bestehenden Katalog zu ergänzen, da immer nur in einem Katalog gleichzeitig gesucht werden kann. Dies erreichen Sie über die Option *Verzeichnis|Editieren|Hinzufügen*.

Dort können Sie dann detailliert angeben, welche Dateien das Produkt beschreiben.

DMI-Browser

Was bitte ist »DMI« (Desktop Management Interface)? Einfach und verständlich, ist DMI in der Lage, nach einem neuen Standard selbstverständlich, Systemkomponenten zu erkennen. Dazu müssen die entsprechenden DMI-Service-Layer vorhanden sein.

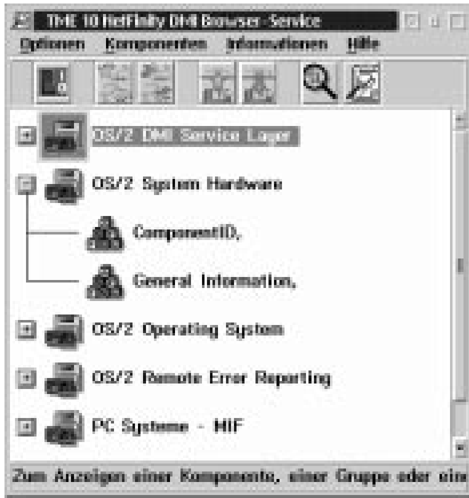


Abb. 10.26: DMI-Service-Layer

Jedem dieser Service-Layer sind wiederum sind einzelne Komponenten zugeordnet, die wiederum einzelne Schlüssel kennen, die Werte eines bestimmten Types annehmen können.

10.1.5 Availability-Funktionen

Was sind Critical Ressources?

Prinzipiell sind alle Ressourcen, die zur täglichen Arbeit absolut notwendig sind, als critical zu betrachten. Wie Sie jedoch sicher bemerkt haben, liegt die letztendliche Entscheidung, wen Sie was wann wie oft sehen lassen wollen, was sie wie oft und in welchen Intervallen protokollieren wollen, bei Ihnen selbst. Also denken Sie an daran, daß Sie außer Systems Management zu betreiben, auf Ihrem PC auch noch arbeiten wollen.

Systemüberwachungsfunktion

Die Systemüberwachungsfunktion bietet Monitore für die vielfältigsten Systemprozesse. Auch hier sind die angebotenen Monitore abhängig von der Installationsumgebung, so werden TCP/IP-Protokollüberwachungen nur angeboten, wenn TCP/IP tatsächlich installiert ist. Einige Monitore sind fix, wie zum Beispiel Prozessor, Prozesse, Threads, Swapdatei, Festplattenplatz frei und belegt,.....

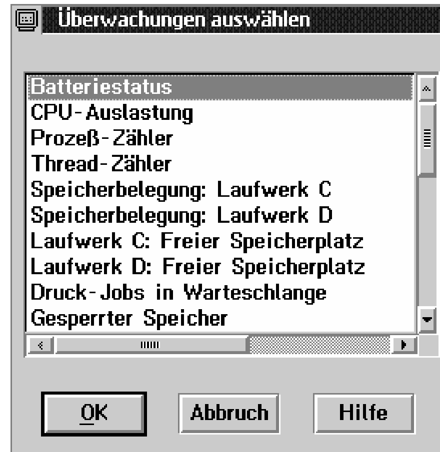


Abb. 10.27: Systemüberwachung – Monitore

Nachdem Sie alle relevanten Monitore ausgewählt haben, können Sie diese beliebig auf der Arbeitsoberfläche anordnen. Die Anzeige der Werte kann als Text, Echtzeitgrafik (mit Protokolldatei) und als Diagramm angezeigt werden. Aus unerfindlichen Gründen werden die Positionen und Anzeigearten der einzelnen Fenster leider nicht gespeichert, so daß bei erneutem Aufruf der Systemüberwachung alle ausgewählten Monitore links unten übereinander gestapelt werden.

Jeder Monitor hat ein Kontextmenü, das mit der rechten Maustaste erreicht werden kann. Dort wird die Darstellungsart, sowie gegebenenfalls durchzuführende Protokolle definiert. Mit *Öffnen|Schwelle* wird ein Fenster zu Einstellung von bestimmten Schwellwerten angezeigt.

Dort können Sie nun neue Schwellwerte definieren, indem Sie den Wert (erstellen) überschreiben und in den Feldern jeweils einen Wert definieren. So können Sie zwischen zu großen Werten (zum Beispiel Swap-Datei) oder zu kleinen Werten (zum Beispiel freier Plattenplatz) unterscheiden. Weiterhin zwischen Fehler und Warnung; diese Unterscheidung, gleichfalls wie die Angabe in Wertigkeit, haben Auswirkungen im Alert-Manager, der noch beschrieben wird. Logischerweise kann ein Schwellwert entweder größer oder kleiner sein. Das mittlere Feld dient dazu, einen Schwellwert als wieder unterschritten zu melden.

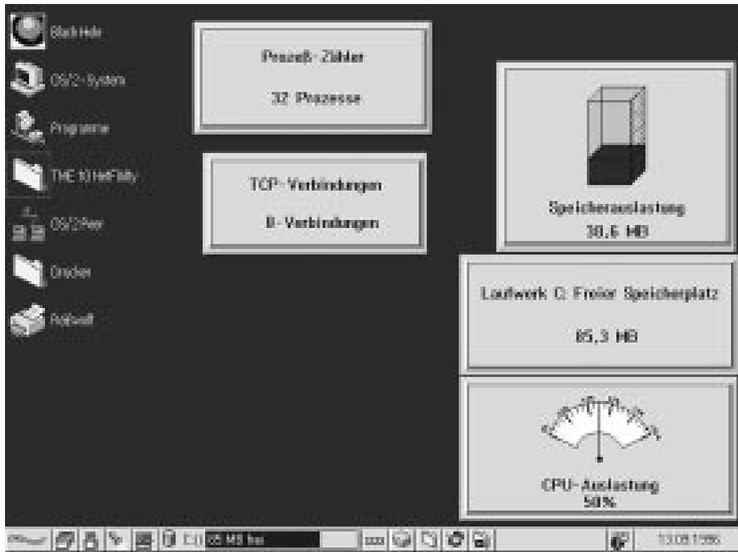


Abb. 10.28: Systemüberwachungs-Anzeigen



Abb. 10.29: Systemüberwachung – Schwellwertdefinition

Wird nun der definierte Schwellwert überschritten, wird ein Alert ausgelöst, der dem Alert-Manager übergeben wird. Dieser kennt nach der Installation nur zwei Reaktionen: Anzeige des Alerts auf dem lokalen Bildschirm und Speicherung des Alerts im Logfile.

ECC Setup

Bei Systemen, die mit ECC, also fehlerkorrigierendem Speicher, ausgestattet sind, finden Sie die Möglichkeit, das Verhalten dieser Speichersteine im Fehlerfall zu konfigurieren.

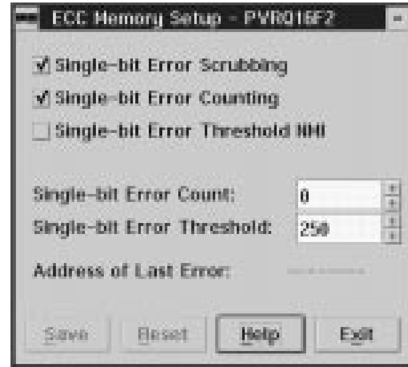


Abb. 10.30: ECC-Memory-Setup

POED

POED steht für »Power ON Error Detect«, die Möglichkeit, von bestimmten Rechnern Konfigurationsänderungen (Batterie leer, Festplatte defekt,...) zu protokollieren. Die ersten PCs taten dies mit der Anzeige von Zahlencodes am Bildschirm, später dann auch mittels Protokolldateien. Mit TME10 Netfinity sind diese PCs in der Lage, Konfigurationsfehler über eine Minimal-Netbios-Implementierung über einen installierten Netzwerkadapter an einen TME10-Netfinity-Manager zu melden, auch wenn der PC nicht in der Lage ist, korrekt zu booten.

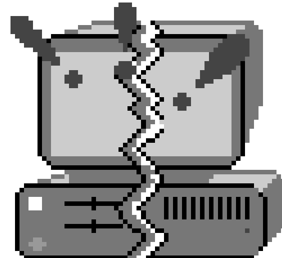


Abb. 10.31: POED-Symbol

Predictive Failure Analysis

Sinngemäß bedeutet »Predicitve Failure Analysis« die Weissagung einer Festplatte, daß Sie demnächst ausfallen wird. Dies ist allerdings mittlerweile zu einem anerkannten und zuverlässigen Verfahren ausgereift, so daß Festplatten, die dieses Verfahren unterstützen, etwa 24 Stunden vor Ausfall den drohenden Ausfall melden können.

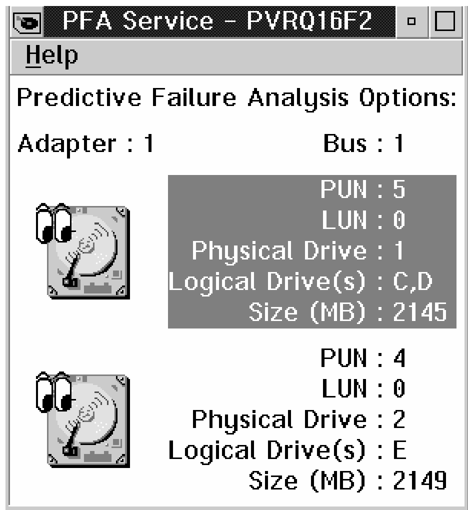


Abb. 10.32: Predictive Failure Analysis – Optionen

Sollte es in der Tat zu einem solchen drohenden Plattenausfall kommen, meldet sich die Festplatte über den Alert-Manager.

Alert Manager

Wie Sie aus den vorausgeganenen Kapitel gesehen haben, ist der Alert-Manager die zentrale Schaltstelle, um auftretende Meldungen, Störungen, Schwellwertüberschreitungen oder Ausfälle zu behandeln. Nach der Installation sind für alle auftretenden Situationen die Aktionen *Anzeige* auf dem lokalen Bildschirm und *Speichern in einer Protokolldatei* vorgegeben. Der Alert-Manager läßt sich jedoch in seinen Aktionen sehr viel weiter ausbauen und viel detaillierter auf bestimmte Vorfälle reagieren.

Über die Auswahl *Profile* bekommen Sie alle definierten Alerts des Alert Managers, die Sie gegebenenfalls Ihren Bedürfnissen anpassen können. Im Regelfall ist dies nicht notwendig. Über den Schalter *Aktionen* können Sie zulässige Aktionen definieren. Haben Sie hierbei einen Alert aus dem darüberliegenden Fenster ausgewählt, gilt die definierte Aktion nur für diesen Fehlerfall. Wählen Sie keinen Alert aus, so gilt Ihre Definition für alle eingehenden Alerts.

Im Feld *Alerttyp* finden Sie die Definition von Schwellwerten in der Systemüberwachung wieder, ebenso wie die Wertigkeit einer Warnung/Fehlermeldung.



Abb. 10.33: Alert Manager-Hauptfenster

Hier können Sie nun definieren, welche Aktionen bei einem Schwellwert mit einer bestimmten Wertigkeit ausgelöst werden soll. Also zum Beispiel soll nur eine Fehlermeldung mit der Wertigkeit 1 auf dem Bildschirm angezeigt werden, eine Fehlermeldung mit der Wertigkeit 3 wird nur im Logfile protokolliert. Hierbei ist zu beachten, daß die Wertigkeit 1 die höchste Priorität besitzt.

Zusätzlich haben Sie die Möglichkeit, eine Anwendungs-ID zu definieren, die angibt, von welcher Servicekomponente von TME10 Netfinity dieser Alert geschickt wurde. Des weiteren können Sie bei einer Netzinstallation angeben, von welchem Sender Sie Alerts entgegennehmen. So können Sie, ohne eine Manager-Version installiert zu haben, von einem oder mehreren bestimmten Systemen Meldungen empfangen und Aktionen daraufhin starten.

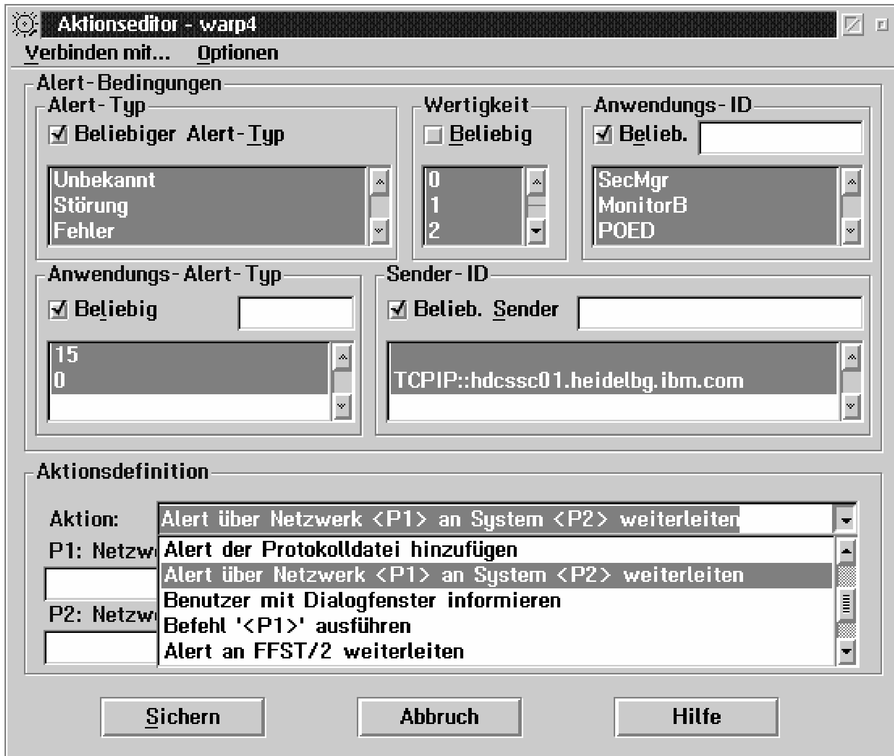


Abb. 10.34: Alert Manager – Aktionseditor

Bei der Aktionsdefinition können Sie nun sehr vielfältige Aktionen definieren:

– Alert an einen Pager / City-Ruf-Empfänger weiterleiten

– Einen Befehl (auch Stapeldatei) im Hintergrund ausführen

– Ein DMI-Ereignis an den emsprechenden DMI-Layer senden

– Alert der Protokolldatei hinzufügen (Standardeinstellung)

– Alert über Netzwerk X and System Y weiterleiten (hier kann X eines der unterstützten Protokolle sein, und Y ein jeweils identifizierender Systemename)

– Benutzer mit Fenster informieren (Standardeinstellung)

– Einen Befehl (auch Stapeldatei) im Vordergrund ausführen (sichtbar)

– Alert an FFST/2 weiterleiten

– Audiodatei (WAV) abspielen

– Numerischen Piepser aktivieren

Wie Sie sehen, sind hier der Automation Tür und Tor geöffnet. So könnte das Löschen der Datei Config.Sys eine Meldung auf dem Bildschirm produzieren, eine WAV-Datei (Polizeisirene) ertönt, mittels eines REXX-Scripts wird aus einem Sicherungsverzeichnis die Datei wiederhergestellt, der Administrator wird bei Email und Piepser informiert. Dies sollte den Benutzer davon überzeugen, diese Aktion das nächste Mal doch wohl besser zu lassen.

10.1.6 Operations-Funktionen

Die meisten der Operations-Funktionen können lediglich über den Sicherheitsmanager für den entfernten Zugriff über ein Netzwerk freigegeben werden. Nur die Überwachung kritischer Dateien kann als lokale Funktion ausgeführt werden. Dennoch werden die übrigen Funktionen erläutert, um zu verdeutlichen, wie diese von einem entfernten System genutzt werden können.

Überwachung kritischer Dateien

Der Name dieser Funktion ist sprechend; alle für Sie wichtige Dateien können Sie hier überwachen. Standardmäßig sind drei Dateien mittels Check-Box auswählbar: »config.sys«, »startup.cmd« und »autoexec.bat«.

Alle übrigen Dateien können über ein Dateiauswahlfenster selektiert werden.

Remote Control

Von einer entfernten Station aus kann mittels Remote Control die gesamte Steuerung des lokalen PCs vorgenommen werden. Die Funktionalität wurde aus dem IBM-Produkt »DCAF« (Distributed Console Acess Facility) übernommen und auf den Transportmechanismus von TME10 Netfinity angepaßt. Die Übernahme eines entfernten PCs kann im passiven oder aktiven Modus geschehen. Beim passiven Modus hat der Benutzer des lokalen PCs weiterhin die Möglichkeit, mit Tastatur und Maus zu arbeiten, die entfernte Station ist nur Beobachter, dies ist zum Beispiel sinnvoll, um einer zentralen Stelle eine Fehlersituation vorzuführen. Im aktiven Modus übernimmt die entfernte Station die gesamte Kontrolle des lokalen PCs. Der Benutzer hat nun Beobachter-Status.

In beiden Modi wird dem lokalen Benutzer mit einem Symbol auf der Arbeitsoberfläche angezeigt, daß eine Remote-Control-Sitzung besteht. Mit der Tastenkombination [Strg]+[T] kann er diese auch beenden.

Remote-Session

Ähnlich wie bei der Remote-Control-Sitzung wird auch hier eine Sitzung mit einem entfernten PC aufgebaut. Der Unterschied ist, daß hier lediglich ein Textfenster besteht. Die Funktionalität ist ähnlich einer Telnet-Sitzung und wird durch ein remotes OS/2-Fenster realisiert. Dort

können nur textbasierende Kommandos ausgeführt werden; wird ein PM-Programm gestartet, so wird dieses auf dem lokalen PC angezeigt, die entfernte Station hat keinen Zugriff darauf.

Screen View

Viele Sharewaretools und Grafikprogramme kennen die Möglichkeit einer Momentaufnahme des Bildschirms mittels einer speziellen Tastenkombination. Auch in TME10 Netfinity haben Sie die Möglichkeit, lokal oder remote einen sogenannten Screenshot durchzuführen und so zum Beispiel Meldungen auf dem Bildschirm zu dokumentieren.

Transfer Files

Hier läßt sich sehr leicht eine Parallele zum ftp aus der TCP/IP-Welt herstellen. Es besteht also die Möglichkeit, Files von einem remoten PC zu holen, bzw. die Files dorthin zu transferieren. Der Unterschied zum TCP/IP-FTP ist einfach: Die Transfer Files-Funktion von TME10 Netfinity basiert auf dem Netfinity-Transport, welcher auch Übertragungen über Netbios, IPX und seriellen Verbindungen zuläßt und nicht auf TCP/IP beschränkt ist.

Update System Programs

In den seltenen Fällen, daß Sie einen PC mit einer Referenzpartition vorfinden (einige wenige IBM-PCs waren damit ausgestattet), haben Sie die Möglichkeit, die Informationen auf dieser Partition von einem entfernten System aus zu aktualisieren.

Bei diesen PCs (und auch jenen, die eine Referenzdiskette brauchten) wird die Hardwarekonfiguration auf einer speziellen Partition auf der Festplatte vorgehalten. Werden neue Adapter oder Komponenten eingebaut, müssen auf diese Partition ADFs (Adapter definition files) hinzugefügt werden. Dies geschieht in der Regel lokal durch ein spezielles Programm, das nur durch Drücken der [Einf]-Taste während des Booten des Rechners erreicht werden kann. Mittels TME10 Netfinity kann nun dieses Files von einer entfernten Station im laufenden Betrieb auf diese Partition eingebracht werden.

RAID-Manager

Sollten Sie über einen Server-PC mit RAID-Festplatten verfügen, finden Sie ein zusätzliches Symbol für die Verwaltung dieser Platten. Auch hier verhält es sich analog zum System-Partition-Access; die Konfiguration dieser Platten wird in der Regel durch eine spezielle Boot-Diskette aktiviert und mittels eines DOS-Programmes durchgeführt. Mit TME10 Netfinity besteht auch hier die Möglichkeit, diese Plattenkonfiguration lokal oder remote im laufenden Betrieb durchzuführen.

11 Das BonusPak – Faxlösung, Textverarbeitung und einiges mehr

von Ina Herbert, Andreas Portele, Bernd Rohrbach und Oliver Mark

Wie auch bei der vorherigen OS/2-Version finden Sie im Lieferumfang von der Version 4 das sogenannte BonusPak. Dabei handelt es sich um eine Reihe von Anwendungsprogrammen, wie z.B. IBM Works oder FaxWorks/2 sowie einige Hilfsprogramme, über die beispielsweise die Installation von Hewlett Packard und Lexmark-Druckern vereinfacht wird.

Das neue BonusPak enthält »IBM Works« als integrierte Anwendung mit Textverarbeitung, Tabellenkalkulation und Datenbank. Für detaillierte Auswertungen ist das Bericht-Modul zu nutzen. Zahlreiche PIM-Tools (Personal Information Manager) helfen Ihnen, Ihre Adressen und Termine besser zu verwalten.

»HyperAccess Lite« für OS/2 ist ein komfortables Terminalprogramm. Es unterstützt Wählverbindungen, z.B. für Mailboxen/Bulletin Boards. Auch das Versenden und Empfangen von Dateien ist darüber möglich.

Der »CompuServe Information Manager« (CIM) für OS/2 ermöglicht Ihnen den Zugriff auf CompuServe-Online-Services.

Über »FaxWorks für OS/2« ist das Senden und Empfangen von Faxmitteilungen möglich.

»VideoIN für OS/2« nutzen Sie für das Abspielen und Speichern digitaler Videosequenzen.

»MarkVision für OS/2« und »HP-JetAdmin« sind Ihnen bei der Installation und Konfiguration von Lexmark- und HP-Druckern behilflich.

»Remote Support für OS/2« ermöglicht die Online-Diagnose durch IBM-Service-Mitarbeiter, wenn Sie mit Ihrem OS/2-System Probleme haben sollten.

»AskPSP« unterstützt Sie bei der Diagnose und Beseitigung von Fehlern. Interaktiv erfragt das System die Symptome, grenzt Fehlerquellen ein und bietet Ihnen auch Lösungsvorschläge an.

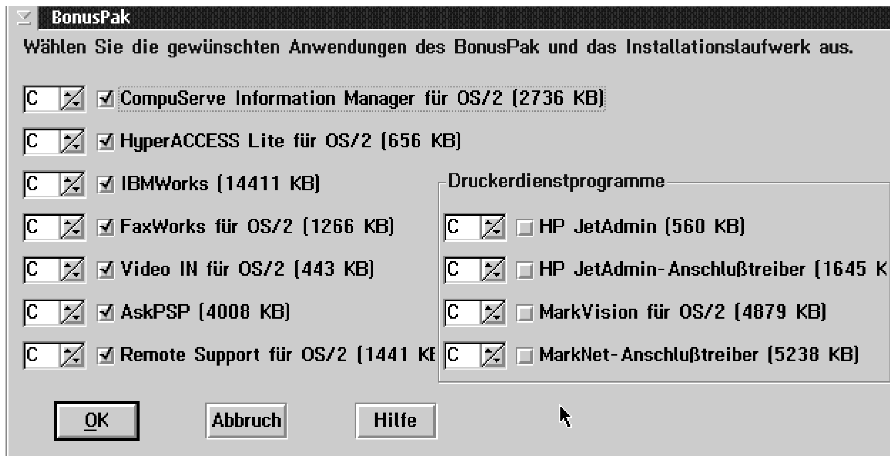


Abb. 11.1: Über das Installationsprogramm von OS/2 sind auch die Anwendungen des BonusPaks zu installieren/deinstallieren

11.1 IBM Works und die PIM-Tools

von Ina Herbert

Für den Nutzen einer integrierten Lösung wie IBM Works ist nicht nur die Breite der angebotenen Funktionalität, sondern auch deren Tiefe entscheidend. IBM Works ist in allen aufgeführten Bereichen durchaus mit der Leistungsfähigkeit von Einzelpaketen vergleichbar.

Ein wesentliches Merkmal dieser Lösung ist die Verfügbarkeit von einmal eingegebenen Daten in allen Modulen. Dadurch entfällt die Mehrfacheingabe identischer Informationen. Wurde beispielsweise eine Adreßdatenbank angelegt, kann diese problemlos ohne jegliche Konvertierung im Textverarbeitungsmodul als auch in den anderen Bereichen genutzt werden. Dadurch ergibt sich auch der geringe Fehleranfall, da Daten nur einmal erfaßt und im Bedarfsfall nur noch ergänzt werden müssen. Durch die leichte Integrierbarkeit wird die Bearbeitungszeit beim Umgang mit Texten, Zahlen und Grafiken minimiert.

IBM Works ist so ausgelegt, daß sie von allen Büromitarbeitern eines Unternehmens genutzt werden kann. Hierzu gehören Sekretärinnen und Sachbearbeiter, aber auch Führungskräfte auf der mittleren und höheren Managementebene. Es stehen Werkzeuge zur Verfügung, die die Handhabung des Produktes vereinfachen und auch die Unterstützung von häufig genutzten Arbeitsschritten bieten.

11.1.1 IBM Works-Textverarbeitung im Mittelpunkt

Im Mittelpunkt von IBM Works steht die Works-Textverarbeitung. Sie beinhaltet alle Merkmale, um den tagtäglichen Schriftverkehr zu vereinfachen. Benötigen Sie ein Schriftstück öfters, kann dieses als Schablone definiert werden. Eine Schablone ist nicht anderes als eine Dokumentenvorlage. Alle definierten Attribute, wie die Schriftgröße und -art sowie der Seitenaufbau werden darin gespeichert. Möchten Sie ein neues Dokument erstellen, brauchen Sie nur diese Schablone zu aktivieren und alle Einstellungen werden übernommen. Sie erreichen, daß alle Dokumente ein konsistentes Erscheinungsbild haben.



Abb. 11.2: Über das Anwendungen-Fenster können Sie neue Dateien anlegen sowie Dateien öffnen

Doch nicht nur das Erstellen von Dokumenten ist mit der Works-Textverarbeitung möglich. Sie können damit auch Serienbriefe erstellen. In der Textverarbeitung ist der Text zu erfassen, der bei allen Briefen identisch sein soll. Die variablen Daten, wie die Anschrift oder die Anrede wird aus der Works Datenbank genommen. Dafür sind lediglich die Felder in dem Dokument zu plazieren. Bei der Druckausgabe der Briefe werden dann die Datenbankinhalt übernommen.

Ein weiteres Merkmal der Works-Textverarbeitung ist, daß Sie problemlos aus IBM-Works-Tabellenkalkulation Arbeitsblätter in Dokumente einfügen können. So ist es einfach, ein Geschäftsbericht mit Umsatzdaten aus der IBM-Works-Tabellenkalkulation zu erstellen. Dabei ist zusätzlich zu vereinbaren, ob eine dynamische Verbindung zwischen dem Rechenblatt der Tabellenkalkulation und der Tabelle in einem Dokument hergestellt werden soll. Ändern sich die Werte in der Tabellenkalkulation, findet in der integrierten Tabelle eine Aktualisierung der Werte statt. Auch können Sie mit einem Doppelklick mit der Maus auf die integrierte Tabelle in dem Dokument die Tabellenkalkulation starten.

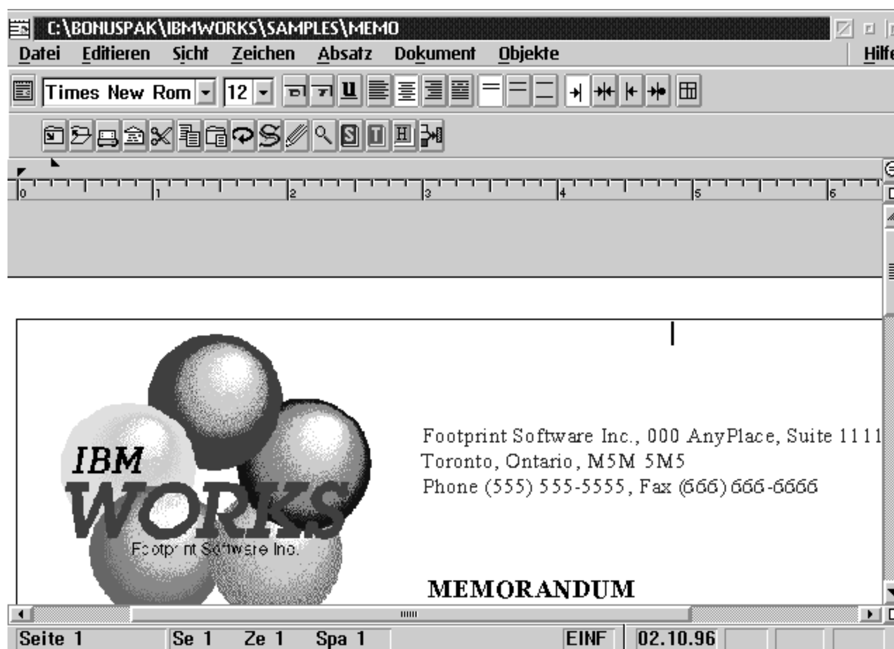


Abb 11.3: Die Benutzeroberfläche der IBM-Works-Textverarbeitung

Um eine verknüpfte Tabelle in dem Dokument zu bearbeiten, brauchen Sie nicht extra das Tabellenkalkulationsprogramm starten und die Tabelle laden. Das Ändern von Zelleninhalten und das Hinzufügen neuer Zellen sowie die Formatierung ist direkt aus dem Textverarbeitungsprogramm realisierbar. Sie haben auf Ihrem Bildschirm das Dokument mit der dynamisch verbundenen Tabelle und machen darauf lediglich einen Doppelklick

Neben dem Einfügen von Works-eigenen Komponenten verfügt die Textverarbeitung auch über Merkmale, um beispielsweise Bilder in Dokumente einzufügen. So kann Ihr Firmenlogo, wenn es als Grafikdatei vorliegt, in ein Dokument integriert werden. Grafiken können direkt in den Fließtext plaziert werden. Detailliert ist dabei zu bestimmen, wie der Textfluß um die Abbildung erfolgen soll. Wahlweise kann ein Rahmen gezogen werden.

Um Informationen übersichtlich in tabellarischer Form darzustellen, ist eine Tabellenfunktion direkt aus der Textverarbeitung zu aktivieren. Und wer geschickt mit der Maus umgehen kann, hat mit der Freihandzeichen-Funktion des Programms weitere Möglichkeiten ein Dokument zu gestalten.

Komfortable Benutzeroberfläche

Der IBM-Works-Textverarbeitungs-Bildschirm gliedert sich in mehrere Bereiche. In der Titelleiste, am oberen Bildschirmrand wird der Name des Dokuments angezeigt. Haben Sie die Datei noch nie gespeichert, wird automatisch der Name *Dokument* mit einer Zahlenangabe

angezeigt. Darunter befindet sich das Hauptmenü. Durch das Anklicken von *Menüoptionen* wird ein Menü geöffnet und weitere Untermenüpunkte stehen zur Verfügung. In der nächsten Leiste werden verschiedene Symbole angezeigt. Auch können Sie hier die Schriftart und deren- größe ändern. Anstatt über das Menü und Untermenüpunkte kann einfach per Mausklick die Darstellung eines Textes von linksbündig auf zentriert geändert werden. Auch können Sie über die Symbole bestimmen, wie der Zeilenabstand, einzeilig oder mehrzeilig, erfolgen soll. Über die nächsten vier Symbole können Sie die Tabulatoren setzen. Dabei wird unterschieden, ob die Ausrichtung für die Tabulatoreneinstellung links, rechts, zentriert oder dezimal erfolgen soll.

Neben der Symbolleiste verfügt die Textverarbeitung über eine Zeichenpalette und eine Darstellungspalette. Möchten Sie in ihrem Dokument Freihandzeichnungen erstellen oder Textpassagen umranden, sind dafür die Zeichenwerkzeuge zu nutzen. Diese Zeichenpalette können Sie, genauso wie die Symbolleiste, im Bedarfsfall aktivieren und deaktivieren. Um einzelne Zeichen oder einen markierten Textabschnitt definierte Attribute zuzuordnen, ist die Darstellungspalette zu nutzen. Dabei handelt es sich um eine Leiste, die im Bedarfsfall aktiviert wird. Auch können Sie damit auch eigene Darstellungsvarianten definieren und speichern und per Mausklick die Formatierungen einen Textbereich zuordnen. Der Einsatz der Darstellungspalette ist dann insbesondere sinnvoll, wenn ein Textbereich mit mehreren gleichen Attributen immer versehen werden soll. Sie ersparen sich so den umständlichen Weg die notwendigen Formatierungen über das Menü oder über die Symbolleiste einzeln auszuwählen.

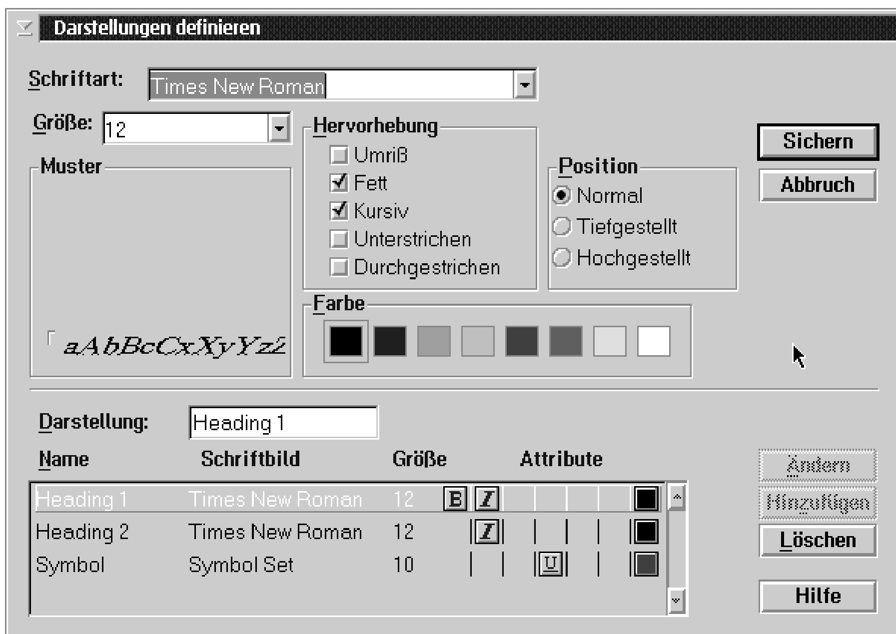


Abb. 11.4: So einfach können Sie neue Darstellungen anlegen

An unteren Bildschirmrand befindet sich die Statusleiste. Auch diese Leiste kann wahlweise ein und ausgeblendet werden. In der Statusleiste wird eingeblendet, wieviel Seiten das Dokument beinhaltet. Auch können Sie erkennen, auf welcher Seite gerade der Cursor ist. Dabei wird die Zeichen- und Spaltenanzahl eingeblendet. Neben dem Datum wird auch angezeigt, ob Sie im EINF- oder ENTF-Modus sind. Dieser Modus können Sie durch Drücken der [Entf]-Taste auf Ihrer Tastatur ändern.

Haben Sie einen Brief geschrieben und möchten ihn in seiner Gesamtheit am Bildschirm betrachten, um beispielsweise die Textaufteilung zu kontrollieren, hat das Textverarbeitungsprogramm dafür den Layout-Modus zu bieten. Über diese Funktion sollten Sie Dokumente, bevor sie über den Drucker ausgegeben werden, betrachten. Die Position von Text, Tabellen und Bildern werden dabei so angezeigt, wie sie beim Ausdruck erscheinen. Zusätzlich können Sie im Layout-Modus die eingestellten Ränder mit der Maus verschieben. Direkt im Layout-Modus kann der Druckvorgang gestartet werden. Sie brauchen dafür also nicht extra in das Hauptmenü verzweigen, sondern durch das Anklicken des Druckersymbols die Ausgabe starten.

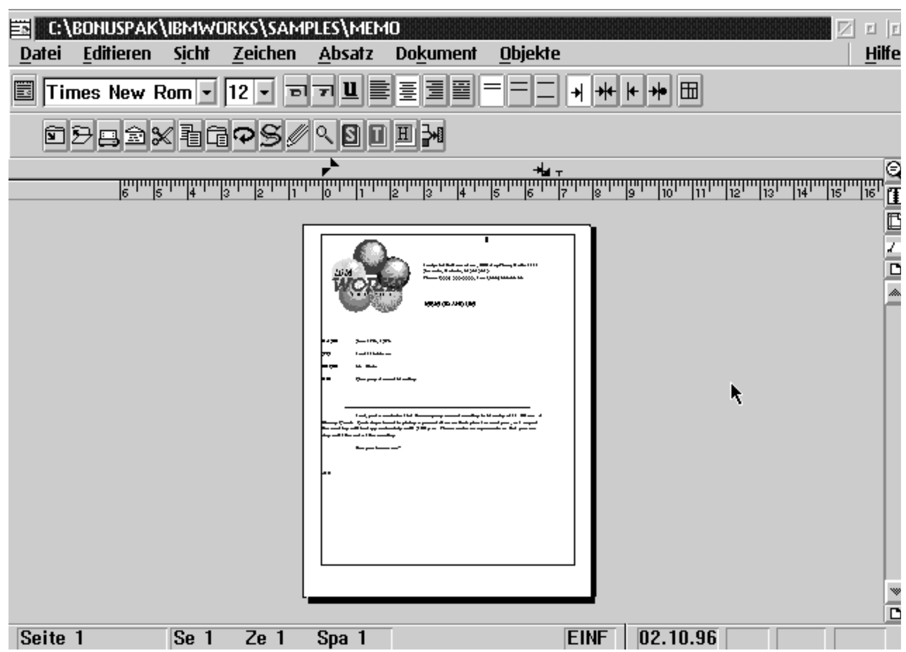


Abb. 11.5: Das Erscheinungsbild eines Dokuments können Sie gut über den Layout-Modus kontrollieren

Neben der Rechtschreibprüfung ist im Lieferumfang der Textverarbeitung auch ein Silbentrennprogramm und ein Synonym-Wörterbuch enthalten. Bei der Silbentrennung können Sie angeben, ob die zu trennenden Wörter angezeigt werden sollen oder ob die Trennung automa-

tisch erfolgen soll. Um Wortwiederholung zu vermeiden, ist das Synonym-Wörterbuch zu nutzen. Für das ausgewählte Wort wird eine Liste von synonymen Begriffen angezeigt. Wahlweise kann es dann ersetzt oder beibehalten bleiben.

11.1.2 IBM Works Datenbank und Tabellenkalkulation

Zum Lieferumfang von IBM Works gehört natürlich auch eine Datenbank. Damit können Sie strukturierte Informationen speichern und organisieren. Eine Datenbank besteht aus Datensätzen. Ein Datensatz beinhaltet die Felder, wo die einzelnen Informationen gespeichert werden.

Bevor Sie also das Programm starten, um eine Datenbank anzulegen, sollten Sie sich gründlich überlegen, welche Informationen darin gespeichert werden sollen, welche Felder benötigt werden und wie groß diese sein müssen.

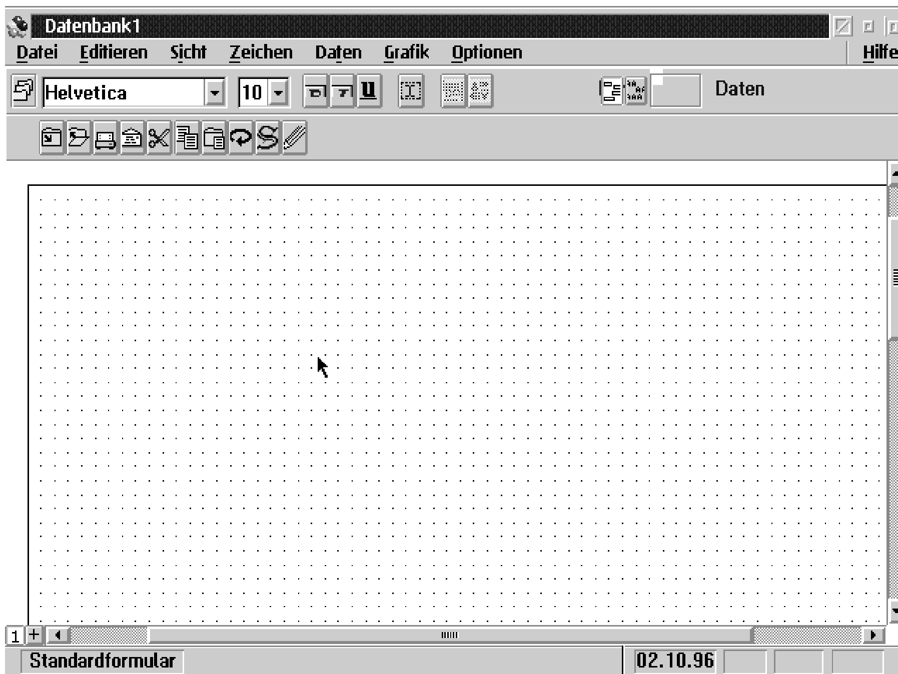


Abb. 11.6: In dem leeren Datenformular müssen Sie die Datenfelder plazieren

Bei der IBM-Works-Datenbank gibt es zwei Modi: das Datenformular und die Dateneingabe. Die Vorgehensweise bei Anlegen einer Datenbank ist etwas gewöhnungsbedürftig. Abwechselnd legen Sie ein Feld an und positionieren dann dieses Feld auf dem Bildschirm in einem sogenannten Formular. Das Formular ist nicht anderes als der noch leere Bildschirm. Über dieses Formular werden dann später die Daten erfaßt.

Beim Anlegen einer Datenbank müssen Sie für jedes Feld einen Namen vergeben. Danach bestimmen Sie die Feldart, d.h., Sie geben an, welche Art von Informationen darin gespeichert werden sollen. Es stehen die Arten *Text*, *mehrzeiliger Text*, *Numerisch*, *Datum*, *Logisch* und *Bild* zur Verfügung. Zusätzlich müssen Sie für jedes Feld angeben, wie groß es sein soll. Ist die Datenbank angelegt und damit alle Felder in dem Formular plaziert, müssen Sie in den Dateneingabemodus wechseln, um Daten zu erfassen. Auch können Sie den Eingabezwang in Felder festlegen sowie Auswahlfelder anlegen.

Datenerfassungsmaske gestalten

Auch bietet das Programm Optionen, eine Datenerfassungsmaske zu gestalten oder ein Datenerfassungsformular zu gestalten. Können Sie geschickt mit der Maus umgehen, haben Sie mit der Freihandzeichen-Funktion des Datenbankprogramms weitere Möglichkeiten, das Formular zu gestalten.

Um Zeichen einfach zu formatieren, ist bei der IBM-Works-Datenbank mit Darstellungspaletten zu arbeiten. Dabei handelt es sich um eine verschiebbare Leiste, über die die Darstellung eines Zeichens oder eines markierten Bereichs einfach geändert werden kann. Sinnvoll ist der Einsatz von Darstellungspaletten, wenn ein Bereich immer mit mehreren Attributen versehen werden sollen. So müssen Sie nacheinander alle Attribute wie Fett, Kursiv und Unterstrichen über das Menü auswählen, sondern können alle Merkmale durch ein Mausklick aktivieren. Auch können Sie eigene Darstellungspaletten definieren.

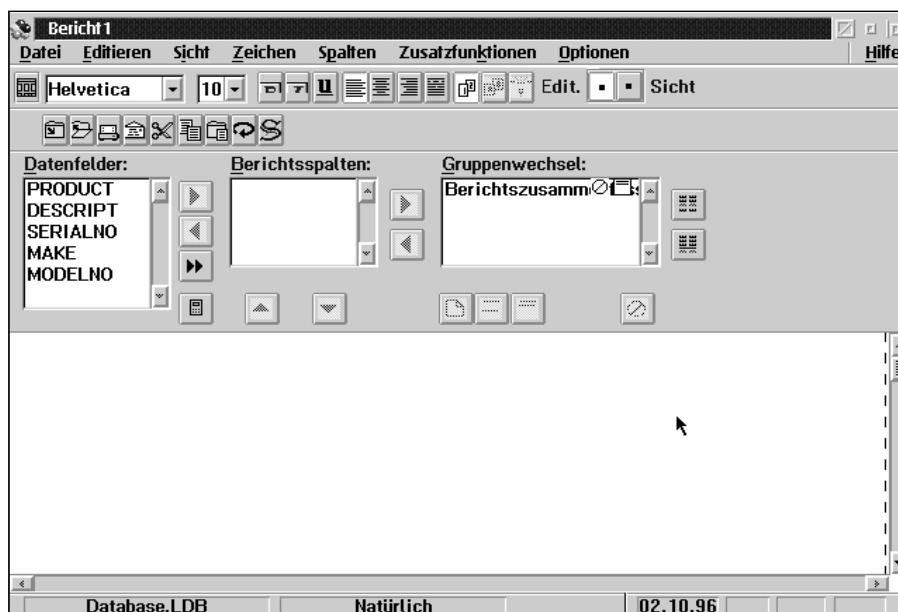


Abb 11.7: Datenerfassungsmasken können Sie individuell gestalten

Datenauswertung über Berichte

In der Datenbanklösung ist die Option *Berichterstellung* integriert. Das bedeutet, selektierte Datensätze aus einer Datenbank werden für die Datenausgabe angeordnet und auch formatiert. Zusätzlich können auch Gruppenwechsel eingestellt werden.

Über das Bericht-Modul können Sie die Daten in den verschiedensten Varianten tabellarisch aufbereiten. Es müssen eine Datenbank und die Felder, die dargestellt werden sollen, bestimmt werden. Neben der Datenausgabe als Liste sind über die Berichtsfunktion zahlreiche Aufwertungen vorzunehmen. Daten können so gruppiert werden. Damit ein Bericht auch optisch gut gestaltet ist, können Felder und Spaltenüberschriften formatiert werden. Berichtstitel, Kopf- und Fußzeilenbereiche sind einzurichten und zu gestalten.

Möchten Sie einen Bericht erstellen oder bearbeiten, müssen Sie im Edit-Modus sein beziehungsweise ihn über diese Schaltfläche aktivieren. Über Sichtmodus können Sie den kompletten Bericht mit allen Formatierungen und Feldinhalten am Bildschirm betrachten.

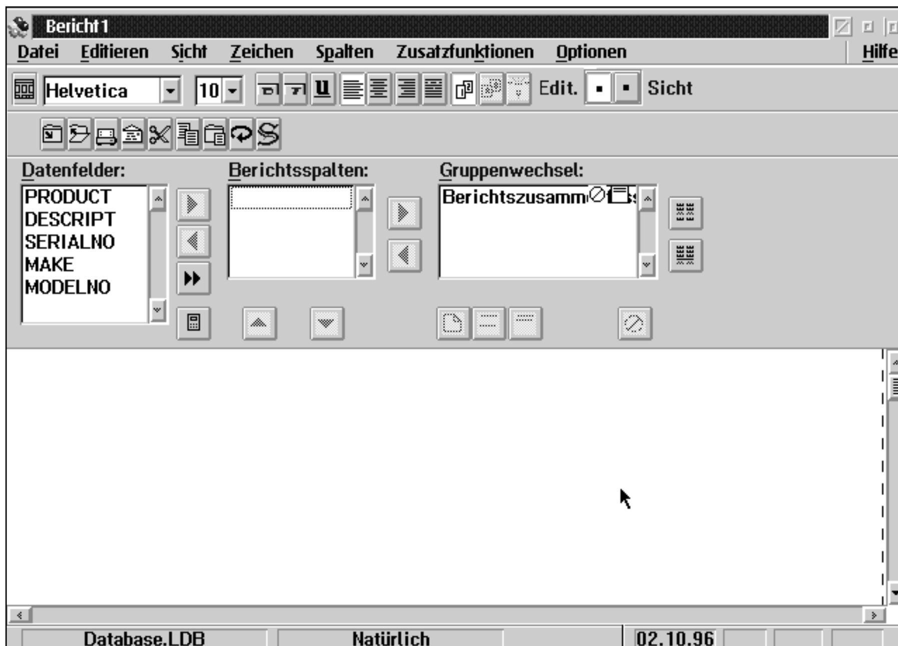


Abb.11.8: Per Mausklick bestimmen Sie, welche Felder in einem Bericht dargestellt werden sollen

Danach werden Ihnen zwei Listen angezeigt. In der ersten Liste mit der Bezeichnung »Datenfelder« finden Sie alle Felder aus der angegebenen Datenbankdatei. In der nächsten Liste mit der Bezeichnung »Berichtsspalten« sind die Felder einzutragen, die in dem Bericht ausgegeben werden sollen.

IBM Works Tabellenkalkulation

Mit der IBM-Works-Tabellenkalulation können Sie nahezu jegliche Arten von Berechnungen durchführen. Dafür stehen Ihnen zahlreiche Funktionen zur Verfügung. So beinhaltet es Textfunktionen, trigonometrische Funktionen, Datumsfunktionen, Uhrzeitfunktionen, statistische Funktionen, Bereichssuchfunktionen, finanzmathematische Funktionen und logische Funktionen. Natürlich können Sie auch beliebige Berechnungsformel direkt eintippen.

Das Arbeitsblatt besteht aus 742 Spalten und 1024 Zeilen, die natürlich nicht alle auf dem Bildschirm sichtbar sind. Um mehrere Ausschnitte einer Tabelle zu betrachten, kann das Fenster auch geteilt werden. So können Sie beispielsweise einfache Zelleninhalte in einen anderen Bereich der Tabelle transferieren.

Um das Datenmaterial entsprechen aufzubereiten, kann es in verschiedenen Diagrammen dargestellt werden. Es stehen Balkendiagramme, Balkenstapeldiagramme, Kurven- und Flächendiagramme zur Verfügung. Gerade dem ungeübten Benutzer kommt der Diagrammassistent zur Hilfe. Er wandelt Zahlenreihen in aussagekräftige Grafiken um. Anhand entsprechender Vorgaben können Sie dabei das Aussehen und die Struktur der Diagramme beeinflussen.

11.1.3 PIM-Module für Ihre Termin- und Aufgabenkoordination

Neben den aufgeführten Schwerpunkten der Workslösung, die Textverarbeitung und Datenverwaltung mit Berichterstellung, finden Sie auch weitere hilfreiche Programme in dem IBM Works-Bereich. Dazu gehört ein Telefon/Adreßbuch, Notizblock, Planer, Aufgabenliste, Jahresplaner sowie ein Terminkalender und ein Ereignismonitor. Diese Programme werden auch als PIM (Personal Information Manager) bezeichnet.

Komfortabel können Sie damit Adressen- und Telefonnummern für den geschäftlichen und privaten Bereich verwalten. Verfügen Sie über ein Modem, sind Teilnehmer aus Ihrem Telefonbuch direkt anzuwählen. Sehr gut lassen sich mit dem Terminkalender des Works-PIM anfallende Termine und auch Aufgaben organisieren. Optimal ist dabei das Zusammenspiel zwischen den PIM-Modulen. Wird ein Termin in den Works-PIM-Terminkalender eingetragen, finden Sie diesen Eintrag auch automatisch in den Modulen *Planer* und in *Liste erledigt*.

Telefon/Adreßbuch

Sowohl für den geschäftlichen als auch privaten Bereich ist das Telefon-/Adreßbuch zu nutzen. Wenn Sie das Programm aktiveren, haben Sie auf Ihrem Bildschirm ein aufgeschlagenes Adreßbuch. Auch die Registerzungen, um direkt zu einem Buchstaben zu wechseln, finden Sie am rechten Rand des »Buchs«. Übersichtlich in Spalten werden die Eintragungen dargestellt. Über Pfeilsymbole ist in dem Adreßbuch seitenweise zu blättern.

Die Einträge werden in Form eines Notizblocks gesichert und in der Kategorie *Private Unterlagen* abgelegt. Dieser Notizblock für private Unterlagen ist als eine Art Ordner zu verstehen. Es setzt sich aus fünf Bildschirmseiten zusammen und beinhaltet alle Informationen zu einem Telefonbucheintrag. So finden Sie auf der ersten Seite ein Reihe von Feldern, wie die Anrede

der Person, deren Name, Vorname, Titel und Firma. Das Anredefeld ist ein sogenanntes Kombinationsfeld. Es darüber eine Liste mit möglichen Anreden zur Verfügung gestellt. Sollte die gewünschte Anrede nicht in der Liste sein, kann sie eingegeben werden. Gleichzeitig wird diese Anrede automatisch mit in die bestehende Liste aufgenommen und kann so für den nächsten Eintrag in das Telefonbuch genutzt werden. Bis zu 20 Anreden können in der Liste des Kombinationsfeldes gespeichert werden. Auch haben Sie zwei Bereiche für die Adresse. Diese können für die Geschäftsadresse und die Privatadresse genutzt werden. Darin sind Angaben wie die Straße, der Ort, das Bundesland, die PLZ und das Land einzutragen.

Abb. 11.9: Übersichtlich gestaltet sich beim Telefonbuch die Datenerfassung

Die Felder im Telefonbuch können maximal bis zu 35 Zeichen aufnehmen und dürften von der Länge her genügen. Das Feld für die Postleitzahl ist 12 Zeichen lang.

In der zweiten Bildschirmseite, die als Sekundär bezeichnet wird, finden Sie das Feld *Symbol*. Dieses Feld enthält den Namen des Symbols, durch das der Eintrag im Telefonbuch dargestellt wird. Dieses Symbol erscheint, wenn das Telefonbuch in der Symbolanzeige dargestellt wird. Es kann benutzt werden, wenn ein Eintrag im Telefonbuch mit der Maus markiert wird und an eine andere Stelle plaziert werden soll. Dieses Markieren und Ziehen des Symbols ist möglich, wenn ein Eintrag aus dem Telefonbuch beispielsweise an das Works-PIM-Modul »Notizblock« oder den Terminkalender übergeben werden soll. Für jeden Eintrag können Sie individuell ein anderes Symbol auswählen. Um diesen Vorgang zu erleichtern, ist ein Symbol-Suchen-Schalter auf der zweiten Bildschirmseite. Die Verzeichnisstruktur auf der Festplatte kann darüber nach Symboldateien durchforstet werden.

In der nächsten Bildschirmseite mit der Bezeichnung »Telefon« werden detailliert alle Angaben für eine reibungslose Kommunikation erfragt. Bis zu acht Telefonnummern pro Person können erfaßt werden. Damit Sie sich nicht an der Telefonnummer orientieren müssen, kann für jede Nummer eine Bezeichnung vergeben werden. Neben den Standardbezeichnung wie Geschäft1 und Privat2, die einfach aus einer Liste auszuwählen sind, sind auch eigene Telefonnummernbezeichnungen zu vergeben. Alle eingegebenen Telefonnummern werden übersichtlich in einer Liste dargestellt und können einfach per Mausklick ausgewählt werden, um sie beispielsweise zu ändern oder zu löschen. Für jede Telefonnummer stehen die Felder Landesvorwahl, Vorwahl, Telefonnummer und Durchwahl zur Verfügung.

Abb: 11.10: Für jede Adresse können mehrere Telefonnummer mit erfaßt werden

Auf den nächsten Seiten des Works-PIM-Telefonbuches gibt es insgesamt zehn Felder mit der Bezeichnung »Benutzerfeld«. Diese können individuell verwendet werden und dienen als Ergänzung, falls die vorherigen Felder für Ihre Eintragungen nicht genügen. Diese Felder sind sogenannte Kombinationsfelder, die der Benutzer wahlfrei mit beispielsweise Stichwörtern füllen kann. Wird ein Kombinationsfeld angeklickt, wird eine Liste mit allen erfaßten Stichwörtern angezeigt. Sie können einfach per Mausklick ausgewählt werden. Die Eintragungen in diese benutzerdefinierten Felder dürfen bis zu 65 Zeichen lang sein. Die letzte Bildschirmseite des Works PIN-Telefonbuches ist für die Aufnahme von Notizen gedacht. Bis zu 512 Zeichen können darin eingegeben werden.

In unteren Bereich des Telefon/Adreßbuches sehen Sie die vier Schaltflächen Anruf, Antwort, Wählen und Protokoll. Verfügen Sie über ein Modem, können Sie über den Anrufschalter einen Teilnehmer aus Ihrem Telefonbuch anwählen. Detailliert wird Ihnen in einem weiteren Fenster das aktuelle Datum und die Uhrzeit angezeigt. Auch läuft eine Stoppuhr, in der die Dauer des Gesprächs ermittelt wird. Zusätzlich können Sie über die vier Optionen *Kontakt herstellen*, *Keine Antwort*, *Nachricht hinterlegen* und *Besetzt* den Status des aktuellen Telefongespräches ermitteln. Ist zu dem ausgewählten Teilnehmer aus Ihrem Telefonbuch keine Verbindung zustandekommen, ist wahlfrei anzugeben, wann nochmals ein Versuch zu starten ist. Diese Angaben werden automatisch in Ihrem Terminkalender eingetragen, der Bestandteil vom IBM Works PIM ist. Sie wechseln also automatisch von dem Telefon/Adreßbuch zum Terminkalender. Darin ist dann auch eine Wecker-Funktion einzuschalten, die an den nächsten Anruf erinnert.

Sie können mit dem Works-PIM-Adreßbuch aber nicht nur Personen anwählen, die dort gespeichert sind. Über eine sogenannte Schnellwahl-Funktion ist das möglich. Sie brauchen nur den Namen und die zu wählende Telefonnummer eingeben, um die Leitung aufzubauen. Neben dem Anrufen von Personen können Sie auch Anrufe entgegennehmen und antworten. Notizen zu den Gesprächen sind wahlfrei einzugeben und auch Folgegespräche einzurichten. Es wird Ihnen das Datum, die Uhrzeit und die Dauer des Gespräches eingeblendet. Die Datums- und Zeitangaben sind schreibgeschützt und können somit nicht geändert werden.

Um einen Überblick über die geführten Telefongespräche zu erhalten, gibt es ein Anrufprotokoll. Dort sind alle Telefongespräche mit dem Datum, die Uhrzeit und Dauer sowie dem Status aufgelistet. Auch die eingegebenen Gesprächsnotizen erscheinen in diesem Protokoll. Wurden sehr viele Verbindungen in dem Protokoll notiert, kann über die Filterfunktion die Anzeige eingeschränkt werden. Dabei ist zu wählen, ob ein Datumsbereich als Kriterium für die Anzeige in der Liste gelten soll oder Namen beziehungsweise ein bestimmter Name.

Eine frei definierte Untermenge der im Telefonbuch gespeicherten Einträge können in die sogenannte Kontaktliste übernommen werden. Es bietet sich an, die Einträge in die Kontaktliste zu übernehmen, die Sie sehr oft benötigen. Sie haben Zugriff auf diese Informationen, ohne das dafür das Telefonbuch aktiviert sein muß. Auch können Sie über diese Liste Änderungen an den Eintragungen vornehmen.

Notizblock

Nahezu jede Art von Informationen können im Works-PIM-Notizbuch erfaßt werden. Neben Text sind darin auch Grafikdateien, die als Bitmap oder Metadatei vorliegen müssen, zu speichern. Die Eingabe von Text oder Grafikdateien erfolgt auf den Notizbuchseiten. Pro Seite kann jeweils bis zu 4 Kbyte Text oder eine Grafik gespeichert werden. Um die Verwaltung der Seiten zu vereinfachen, gibt es ein Inhaltsverzeichnis.



Abb. 11.11: Jegliche Informationen können Sie mit dem Notizblock speichern

Wenn Sie das Works-PIM-Notizbuch öffnen, sehen Sie als erstes ein Inhaltsverzeichnis. Sie müssen dann für die neu zu erstellende Seite/n einen Titel eingeben. Bis zu 64 Zeichen darf er lang sein. Dieser Titel erscheint dann in der Titelleiste am oberen Bildschirmrand der Seite und auch in dem Inhaltsverzeichnis. Handelt es sich um sehr wichtige Notizen oder welche die Sie häufig wieder einsehen und ändern möchten, können Sie zusätzlich zum Eintrag in das Inhaltsverzeichnis eine Registerzunge definieren. So brauchen Sie die Notiz nicht im Inhaltsverzeichnis suchen, sondern können direkt die Seite/n über das Register aufschlagen.

Jederzeit sind die Seiten im Notizblock anders anzuordnen. Über die Maus können die Einträge in dem Inhaltsverzeichnis an eine andere Positionen gezogen werden. Auch ist über das Markieren und Ziehen eines Eintrags mit der Maus auf das Druckersymbol die Ausgabe zu aktivieren.

Terminkalender

Mit den PIM-Terminkalender können Sie gut eine Tages-, Wochen- und Monatsplanung realisieren. Doch nicht nur Termine sind damit zu koordinieren, sondern auch Aufgaben. Wenn Sie das Programm aktivieren, sehen Sie einen aufgeschlagenen Terminkalender, in dem Sie über Pfeilsymbole blättern können. Der aktuelle Tage ist rot markiert

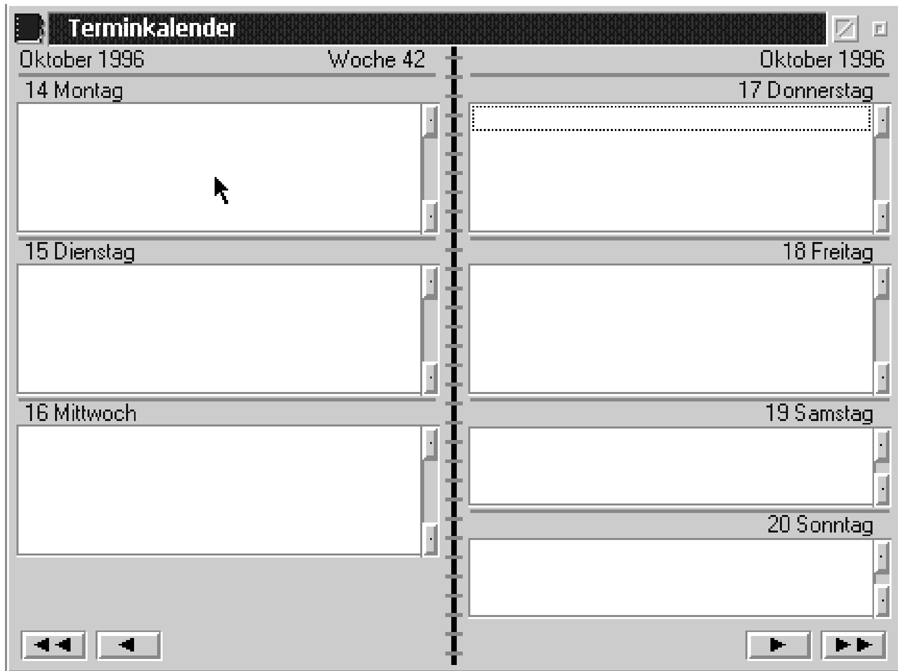


Abb. 11.12: Übersichtlich ist der Terminkalender

Der Works-PIM-Terminkalender stellt vier Anzeige-Modi zur Auswahl. So können Sie zwischen einer Tages-, Arbeitswochen-, normale Wochen- und 14-Tage-Anzeige wählen. Bei der Tagesanzeige stehen Ihnen pro Seite ein Tag zur Verfügung. Eine Arbeitswoche wird auf zwei Seiten dargestellt, wobei der Bereich für den Samstag und Sonntag sehr klein ausfällt. Dieser Bereich wird bei der normalen Wochenanzeige, wie auch die anderen Wochentage gleich groß angezeigt. Bei der 14-Tage-Anzeige haben Sie schon Schwierigkeiten einen Überblick zu bekommen, denn auf zwei Seiten des Terminkalenders sind 14 Tage untergebracht.

Für alle Termine, die Sie in den Kalender eintragen, müssen Sie einen Zeitraum festlegen. Es ist also der Anfang und das Ende des Termins anzugeben. Dazu wird dann die Dauer des Termins eingeblendet. Dabei überprüft das Programm, daß es nicht zu Terminüberschneidungen kommt. Tragen Sie beispielsweise einen Termin von 8:00 Uhr bis 10:00 Uhr ein, steht die Zeit dazwischen für Termineingaben nicht mehr zur Verfügung.

Abb: 11.13: Detailliert ist die Erfassung neuer Termine

Neben einmaligen Terminen sind mit dem Terminkalender auch wiederkehrende Termine zu verwalten. So können Sie beispielsweise angeben, daß Sie Montags um 12:00 Uhr eine Besprechung haben. Detailliert bestimmen Sie, wie oft sich der Termin wiederholt und ob der Intervall auf täglich, wöchentlich oder monatlich einzustellen ist. Auch ist über ein Datumsfeld der Endezeitpunkt anzugeben. Wiederkehrende Termine können auch geändert werden. Dabei ist anzugeben, ob sich die Änderung nur auf einen wiederkehrenden Termin, auf alle oder alle zukünftigen Termine bezieht. Haben Sie beispielsweise am Montag um 19:00 den wiederkehrenden Termin »Turnen in Oberau«, und dieser Termin fällt einmalig aus, können

Sie dies in den Terminkalender eingeben, wobei die weiteren wiederkehrenden Termine davon nicht betroffen sind.

Damit Sie einen Termin nicht vergessen, können Sie sich daran erinnern lassen. Dafür ist die Wecher-Funktion des Terminkalenders zu nutzen. Wahlfrei ist eine Vorlaufzeit einzugeben. Über eine Schiebeleiste können Sie bis zu 2 Stunden Vorlaufzeit einstellen. In Intervallen werden Sie dann an den Termin erinnert. Der Wecker-Intervall ist standardmäßig auf alle 5 Minuten eingestellt.

Mit dem Kalender können Sie nicht nur Ihre Termin erfassen und koordinieren, sondern auch Aufgaben. Es ist der Tag zu wählen, an dem diese Aufgabe erledigt werden soll, sowie ein erklärender Text in den Terminkalender einzutragen. Dieser Text darf bis zu 64 Zeichen lang sein. Der Aufgabe ist dann auch eine Priorität zuzuordnen. Anstatt über die Prioritätsstufen kann wahlweise auch ein Datum für die Priorität eingetragen werden. Und zusätzlich können Sie, wie bei der Terminplanung, die Aufgabe wiederkehrend definieren.

Ein besonderer Schmankerl des Works-PIM-Terminkalenders ist das Starten von Anwendungsprogrammen. Sie können in dem Terminkalender angeben, wann welches Programm aktiviert werden soll. Vergessen Sie beispielsweise immer, nach Arbeitsschluß um 19:00 Uhr eine Datensicherung durchzuführen, können Sie diesen Vorgang nun automatisieren. Pünktlich um 19:00 Uhr wird das Programm zur Datensicherung gestartet.



Abb: 11.14: Über den Terminkalender können Programme ausgeführt werden

Sehr gut ist das Zusammenspiel zwischen dem Terminkalender, dem Planer, Kalender und dem Adreß/Telefonbuch. Wird ein Termin in den Kalender eingetragen, finden Sie im Planer und Kalender dazu einen Vermerk. Im PIM-Works-Modul »Liste Erledigen« wird automatisch ein Eintrag vorgenommen, wenn im Terminkalender eine Aufgabe eingetragen wurde. Auch können Daten aus dem Telefon/Adreßbuch in den Terminkalender übernommen werden.

Liste Erledigt

Um Aufgaben zu notieren und zu verwalten ist das PIM-Works-Modul »Liste Erledigt« zu nutzen. Über ein Fenster mit der Bezeichnung *Starter für Liste erledigen* werden Ihnen über Schaltflächen die Tage des Monats angezeigt. Ist für den aktuellen Tag eine Aufgabe zu erledi-

gen, öffnet sich automatisch die Tagesliste mit den Eintrag. Neben der Prioritätstufe, die von Null bis fünf reichen kann, wird der Aufgabentext angezeigt, was zu erledigen ist.

Soll eine neue Aufgabe definiert werden, ist der Tag auszuwählen, wann diese Aufgabe erledigt werden muß. Sie haben über 60 Zeichen zur Verfügung, um einzugeben, worin diese Aufgabe besteht. Zusätzlich ist anzugeben, ob es sich um eine wiederkehrende Aufgabe handelt.

Damit Sie eine Aufgabe nicht vergessen, können Sie sich über die Wecker-Funktion daran erinnern lassen. Wie auch beim Terminkalender ist wahlfrei eine Vorlaufzeit einzustellen sowie der Weckintervall.

Planer und Jahresplaner

Über den Works PIM-Planer können Sie sich einen sehr guten Überblick verschaffen. In einer tabellarischen Monatsübersicht werden im oberen Bereich des Fensters die Wochentage mit Datum aufgelistet und am linken Fensterrand sehen Sie die Zeitangaben. Alle Termine, Aufgaben auch mit der Wecker-Option gekoppelt und definierte Programmstarts werden darin dargestellt. Zusätzlich können Sie auf dem Planer erkennen, wenn sogenannte Konflikte vorliegen, also sich Termine überschneiden. Alle Eintragungen werden unterschiedlich farbig dargestellt und sind somit auf einen Blick zu erfassen. Zur besseren Orientierung finden Sie am unteren Fensterrand des Planers eine kleine Farbpalette mit Legende. Um zu sehen, was sich beispielsweise hinter einer Terminmarkierung verbirgt, ist lediglich der Mauszeiger darauf zu plazieren und die linke Maustaste zu drücken. Es wird dann beispielsweise der Hinweis eingeblendet »9:00 Termin beim Zahnarzt«.

Doch der Planer verschafft Ihnen nicht nur den Monatsüberblick, sondern Sie können aus diesem PIM-Modul heraus den Terminkalender starten, um neue Termine einzutragen. Das ist ebenso mit der Aufgabeverwaltung »Erledigt Liste« möglich.

Nur zur Anzeige ist der Jahresplaner gedacht. In einem Fenster werden alle Kalenderdaten angezeigt, wobei der aktuelle Tag farbig hervorgehoben wird. Diesen können Sie anklicken, damit die Aufgaben-Liste geöffnet wird.

Ereignismonitor

Über den Ereignismonitor werden Ereignisse wie Wecktermine oder Programmstarts verwaltet Das Modul wird automatisch aufgerufen, wenn Sie ein Works-PIM Modul öffnen und beispielsweise die Wecker-Funktion nutzen. Ist der Ereignismonitor aktiv, wird er ebenfalls in der Fensterliste der aktiven Programm angezeigt.

PIM-Einstellungen

Über das Start- und Vorgaben-Menü des Telefonbuches können Sie Vorgaben für die Works-PIM-Module eintragen. Wenn Sie das Symbol öffnen, sehen Sie am rechten Rand des Fensters die Register Feiertage, Telefonbuch, Termine und Allgemein. Auf der Seite *Feiertage* können Sie eigene Feiertagskategorien einrichten und diesen eine Farbe zuordnen. Bis zu acht Feiertagstypen sind verfügbar. Bereits voreingestellt sind z.B. Geburtstag, Urlaub und Jubiläum. Sie können diese Vorgaben aber ändern. Am unteren Fensterrand ist die Farbpalette, in der Sie

den Feiertagstypen eine Farbe zuordnen können. Diese Farbmarkierung erscheint dann in den Modulen »Terminkalender«, »Kalender«, »Planer« und in der »Liste Erledigen«.

Auf der nächsten Seite des PIM-Einstellungmenüs, im Register *Telefonbuch* sind die Modemeinstellungen vorzunehmen, damit die Datenübertragung (Telefonieren genannt) aus dem Modul Telefon/Adreßbuch möglich ist. Sie müssen den Kommunikationsanschluß wie Com1 oder Com2 bestimmen, an dem Ihr Modem an Ihren PC angeschlossen ist. Auch die Übertragungsgeschwindigkeit bei der Datenübertragung ist anzugeben. Zusätzlich ist dann auch das Impulswahl- oder Mehrfrequenz-Wahlverfahren auszuwählen.

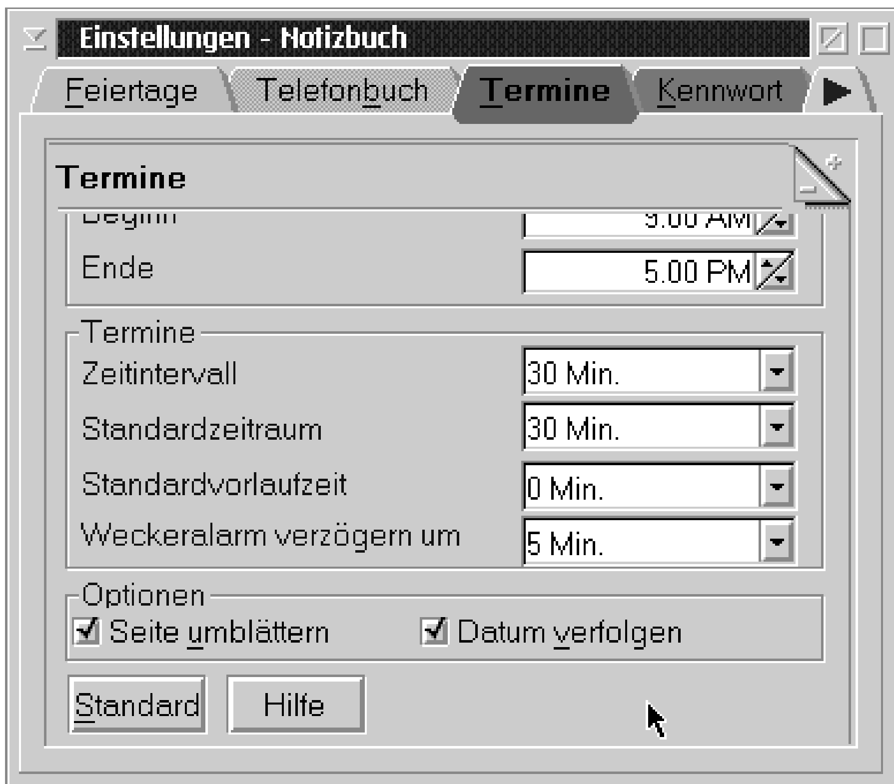


Abb. 11.15: Für alle PIM-Module können Sie über das Einstellungen-Fenster Ihre Vorgaben machen

Auf der nächsten Seite im Register »Telefon« können Sie für die bereits erwähnten benutzerdefinierten Eingabefelder des Telefonbuches Vorgaben eintippen.

Weitere Einstellungen für den Terminkalender sind im Register »Termine« vorzunehmen. Diese Vorgabe wird für die Zeiteinteilung des Terminkalenders benötigt. Auch können Sie hier das Zeitintervall bestimmen. Standardmäßig ist 30 Minuten vorgegeben.

Im letzten Register »Allgemein« sind allgemeine Vorgaben festzulegen. Dazu gehört, ob das Löschen von Einträgen bestätigt werden soll oder ob der Ereignismonitor automatisch zu aktivieren ist.

11.2 HyperAccess Lite

von Ina Herbert

Hyper Access Lite ist ein Terminalprogramm, mit dem Sie Dateien versenden und empfangen können. Es ist auch sehr gut zu nutzen, um sich in die als Mailboxen bekannte Bulletin Boards Systems (BBS) einzuwählen. Die Vielzahl von angebotenen Mailboxen sowie deren unkomplizierte Handhabung ist gerade für den DFÜ-Neuling geeignet. Den Benutzern der Boxen stehen neben Nachrichten- auch Diskussionsforen zur Verfügung. Zusätzlich finden Sie in Mailboxen zahlreiche Shareware-, Freeware- und Public-Domain-Programme zur Auswahl. Diese können Sie auf Ihren PC herunterladen.



Abb. 11.16: Das Telefonbuch bildet den Mittelpunkt von HyperAccess

Wenn Sie HyperAccess Lite aktivieren, ist sofort das Telefonbuch geöffnet, in dem bereits zahlreiche Verbindungsprofile zu beispielsweise ATT Mail, CompuServe und Genie vorbereitet sind. Nach der Konfiguration des Programms können diese definierten Verbindungen per Mausklick aktiviert werden. Jederzeit können Sie das Adreßbuch auch um eigene Verbindungsprofile erweitern. Wahlweise werden diese Einträge durch ein Symbol am Bildschirm dargestellt oder über die Detailanzeige des Programms in tabellarischer Form.

Haben Sie eine Verbindung aufgebaut, können Sie sich mit HyperAccess Lite durch die Mailbox bewegen. Damit Sie Informationen, die während einer Verbindung am Bildschirm angezeigt werden, auch später in Ruhe lesen können, ohne daß dafür Verbindungskosten entstehen, sind diese in eine Datei zu speichern. Beim Senden und Empfangen von Dateien werden die Protokolle Zmodem, 1K Xmodem, Ymodem, Ymodem-G und Kermit Text unterstützt. Neben der Kommunikation mit den genannten Diensten können auch PC-zu-PC-Verbindungen definiert werden. Unter der Bezeichnung *Direkt Verbindung* ist das Verbindungsprofil bereits vorbereitet.

HyperAccess Lite ist zwar eine abgemagerte Variante der Vollversion HyperAccess, dürfte für den normalen DFÜ-Gebrauch aber genügen. Die Einschränkungen von HyperAccess Lite gegenüber der Vollversion betreffen das DFÜ-Profil, das beispielsweise über die Scriptspache der Vollversion eigene Prozeduren erstellen möchte. Auch bietet die Vollversion ein Virenschutzprogramm, das beim Herunterladen von Software zum Einsatz kommt. Neben weiteren Terminalemulatoren werden weitere Dateiübertragungsprokolle und auch Modems unterstützt.

11.2.1 Einrichten von Verbindungsprofilen

Nachdem Sie das Programm installiert haben, brauchen Sie keine aufwendigen Programmkonfigurationen vornehmen.

Damit Sie Dateien verschicken können, müssen Sie als erstes ein Verbindungsprofil definieren. Darin geben Sie neben der Telefonnummer, die Baudrate des Modems, den Anschlußnamen sowie den Modemtyp an. Wenn Sie erst einmal ein Verbindungsprofil angelegt haben, werden diese Daten, bis auf die Telefonnummer der Gegenstelle, automatisch übernommen. Ist zu einer Gegenstelle öfters eine Verbindung aufzubauen, kann der Eintrag in dem Telefonbuch gespeichert werden. Per Mausklick wird dann die Verbindung aufgebaut.

Automatisch wird nach dem Laden des Programms das Telefonbuch geöffnet. Hier sehen Sie in Form von Symbolen bereits einige vordefinierte Verbindungsprofile. Um ein neues Profil anzulegen, ist das Dateimenü zu öffnen sowie *Neu* auszuwählen.



Abb. 11.17: Für ein neues Verbindungsprofil ist ein Systemname einzugeben sowie ein Symbol auszuwählen

Vergeben Sie nun eine Bezeichnung für die neu zu definierende Verbindung. Auch ist ein Symbol, das diese Verbindung im Telefonbuch darstellt, auszuwählen. Über den horizontalen Rollbalken können Sie sich die zur Verfügung stehenden Symbole anschauen und per Mausklick eins auswählen.

Nachdem Sie den Ok-Schalter angeklickt haben, öffnet sich das Fenster *Notizblock Eigenschaften*. Am rechten Bildschirmrand sehen Sie ein Register mit den Bezeichnungen *Beschreibung*, *Kommunikation, ASCII-Einstellungen*, *Terminal-Einstellungen* und *Schriftarten*. Die Seite *Kommunikation* des Notizbuches ist aufgeschlagen. Geben Sie nun die Telefonnummer für die Verbindung ein. Des weiteren ist die Baudrate des Modems, die Schnittstelle und der Modemtyp auszuwählen. Klicken Sie dann den Schalter *Modem-Einstellungen* an. Sie können nun die Anzahl der Wahlversuche und auch das Wahlverfahren Puls oder Ton bestimmen. Diese technischen Einstellungen gelten dann für alle Verbindungen.

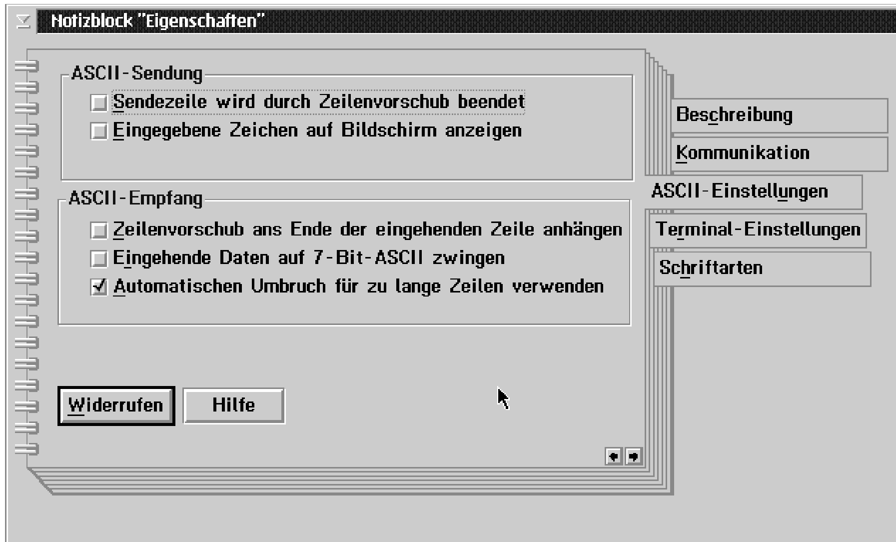


Abb. 11.18: In den Registern sind die Verbindungseinstellungen vorzugeben

Klicken Sie nun das Register *ASCII-Einstellungen* an. Die Einstellungen, die Sie hier vornehmen, betreffen die normale Bildschirminteraktion mit einem entfernten Host. Sie haben keinen Einfluß auf die Dateiübertragungsprotokolle. Sowohl für den ASCII-Empfang und ASCII-Versand können Sie beispielsweise bestimmen, ob die Sendezeile durch einen Zeilenvorschub beendet wird oder ob beim Empfang ein Zeilenvorschub an das Ende einer Zeile angehängt werden soll.

Im Register *Terminal-Einstellungen* können Sie den Terminal-Emulator und deren Einstellungen modifizieren. HyperAccess Lite bietet die Terminalemulatoren ANSI, VT100 und +sowie die Funktion *Automatisch festlegen*. Über diese Funktion kann das Programm die beste Emulation feststellen, die bei der erstmaligen Verbindungsaufnahme eingesetzt wird. Zusätzlich ist auch anzugeben, ob die Funktionstasten-, Pfeil und [Strg]-Taste als Terminaltasten oder Systemtasten dienen sollen.



Abb. 11.19: Beim erstmaligen Verbindungsaufbau kann automatisch die Terminalemulation erkannt werden

Über das letzte Register *Schriftarten* können Sie sowohl die Schriftart als deren Stil und Attribute bestimmen. Diese Angaben können sich auf die Bildschirm- oder Druckerausgabe beziehen.

Nachdem Sie alle Einstellungen überprüft haben, ist aus dem System-Menü *Schließen* zu aktivieren. Wenn Sie nun die Titelleiste von HyperAccess Lite betrachten, wird Ihnen der Systemname der neu definierten Verbindung eingeblendet. Diese Verbindung ist allerdings noch nicht gesichert.

Über das Dateimenü und der *Sichern unter*-Option werden die Einstellungen gespeichert. Automatisch wird Ihnen der Dateiname, der auch gleichzeitig der Systemname der Verbindung ist, vorgeschlagen. Wahlweise können Sie diesen ändern.

Möchten Sie nachträglich an den vorgenommen Einstellungen Änderungen vornehmen, ist nur das *Eigenschaften*-Menü zu öffnen.

Damit das neu angelegte Verbindungsprofil auch sichtbar im Telefonbuch als Symbol dargestellt wird, müssen Sie es im Dateimenü *Telefonbuch öffnen* aktivieren. Klicken Sie dann im *Telefonbuch*-Fenster das *System*-Menü in der oberen linken Ecke an sowie die Option *Sofort aktualisieren*. Sofort wird das Symbol des neu definierten Verbindungsprofils angezeigt.

11.2.2 Verbindung aufbauen

Mit HyperAccess Lite haben Sie zwei Möglichkeiten, eine Verbindung aufzubauen. Der komfortabelste Weg ist der über das Telefonbuch. Machen Sie mit der Maus auf das neu definierte Verbindungssymbol einen Doppelklick, und der Initialisierungs- und Wählvorgang wird aktiviert.

Die andere Variante, eine Verbindung aufzubauen, ist über das Menü von HyperAccess Lite möglich. Im Dateimenü wählen Sie über *Öffnen* wählen aus einer Liste den Dateinamen aus. Danach müssen Sie wiederum aus dem Dateimenü *Verbinden* anklicken. Beim Verbindungsaufbau öffnet sich ein Statusfenster. Um die Verbindung zu trennen, ist die gleichnamige Option aus dem Dateimenü zu aktivieren.

11.2.3 Datei versenden

Damit Sie eine Datei versenden können, müssen Sie zunächst eine Verbindung aufbauen. Danach müssen Sie sich entsprechend den Vorgaben des anderen Systems einloggen. Öffnen Sie dann in HyperAccess Lite *Übertragen* und wählen Sie *Senden*. Geben Sie nun direkt den Dateinamen ein, oder klicken Sie den Schalter *Durchsuchen* an. Wählen Sie dann das Verzeichnis, in dem die zu sendende Datei gespeichert ist, und markieren diese mit der Maus.



Abb. 11.20: Beim Senden einer Datei ist auch das Prokoll anzugeben

Zusätzlich müssen Sie auch das Protokoll angeben. Über den *Senden*-Schalter starten Sie die Übertragung. Nach erfolgreicher Dateiübertragung ist die Verbindung über das Dateimenü zu trennen.

11.2.4 Datei empfangen

Sie haben erfolgreich eine Verbindung aufgebaut und sich auch in dem anderen System eingeloggt, so daß Sie Dateien laden können. Im Übertragen-Menü von HyperAccess Lite wählen

Sie dazu *Empfangen*. Geben Sie das Verzeichnis an, in dem die Datei auf Ihrem System gespeichert werden soll. Wahlfrei können Sie auch einen anderen Dateinamen bestimmen. Nachdem Sie nun noch das Protokoll ausgewählt haben, ist der *Empfangen*-Schalter anzuklicken. Es öffnet sich ein Statusfenster, in dem Sie über den Fortschritt der Übertragung visuell sowie in Kbyte-Angaben informiert werden. Über den *Abbruch*-Schalter kann der Empfang von Dateien jederzeit unterbrochen werden.

11.3 CompuServe Information Manager for OS/2

von Ina Herbert

Man kann wohl CompuServe als die größte Mailbox der Welt bezeichnen. Eine Vielzahl von Angeboten stellt es bereit, die allerdings auch nahezu alle ihren Preis haben. Bereits in der vorherigen Version des BonusPaks war der CompuServe Information Manager integriert. Über Symbole stellt der CompuServe Information Manager die unterschiedlichsten Services bereit. Dabei reicht die Palette von Shopping, Ausbildung, Gesundheit bis über Computer und Gesundheit. Die verschiedensten Foren dienen dem Gedanken- und Erfahrungsaustausch. Selbst an Spielen wie Schach können Sie sich über CompuServe beteiligen.

Sie können jedoch nicht nur auf Informationsrecherche gehen, sondern auch Software über CompuServe laden. Jeglichen Komfort stellt der Information Manager für die Datenübertragung zur Verfügung. Neben Share- und Freeware-Programmen sind auch kommerzielle Lösungen zu beziehen. Allerdings müssen Sie beim Übertragen der Software bedenken, daß im Gegensatz zur Datenübertragung aus dem Internet zusätzliche Kosten entstehen.

Mit dem CompuServe Manager können Sie sowohl Emails erstellen und verschicken sowie auch empfangen. Die Nachrichten können erst verfaßt werden, und dann ist die Verbindung zu CompuServe aufzubauen. Damit an andere CompuServe-Teilnehmer Nachrichten verschickt werden können, benötigen Sie deren Namen und User-ID. Über eine Suchfunktion kann gezielt nach einer User-ID recherchiert werden.



Abb. 11.21: Der CompuServe Information Manager

Natürlich ist die Email-Korrespondenz nicht auf CompuServe-Teilnehmer begrenzt. Einfach sind Nachrichten an Internet oder auch MCI-Teilnehmer zu schicken.

Damit Sie die Adressen von Teilnehmern nicht immer aufs neue manuell eintippen müssen, verfügt der CompuServe Information Manager über ein Adreßbuch. Wahlweise können die Angaben direkt in das Adreßbuch eingetippt werden oder Sie geben an, wenn Sie eine Mail erstellen, ob die Adresse gleich in das Adreßbuch übernommen werden soll. Um gerade große Adressenbestände besser zu organisieren, können auch Gruppen gebildet werden. Und natürlich kann detailliert nach Einträgen gesucht werden.

11.3.1 Elektronische Anmeldung für zukünftige Mitglieder

Hinweis: In den Unterlagen zu OS/2 Warp finden Sie einen Bestellschein, in dem Sie ein kostenloses CompuServe-Einführungsangebot nutzen können. Darin enthalten ist eine Benutzungsgutschrift.

Wenn Sie den CompuServe-Ordner öffnen, haben Sie Zugang zu CIM for OS/2 und auch Signup. Sind Sie bereits CompuServe-Nutzer und haben damit einen Account, können Sie gleich mit CIM for OS/2 loslegen. Ansonsten müssen Sie die Prozedur der Registrierung über *Signup* starten.

Um die Registrierung über das Member Signup durchzuführen, wählen Sie aus dem Signup-Menü die erste Option aus. An dieser Stelle werden Sie aufgefordert, die Registrierungsnummer einzutippen. Haben Sie einen CompuServe-Gutschein geordert, dann müssen im Fenster »Signup-Billing/Country« die Agreement Number eingeben. Ansonsten lassen Sie die Voreinstellung OEM OS2CIM stehen. Auch entnehmen Sie den Gutscheinunterlagen die Seriennummer, die Sie eintragen müssen. Als nächstes ist das Country aus der Liste auszuwählen. Wollen Sie CompuServe privat oder geschäftlich nutzen, ist entsprechend *Personal* oder *Business Account* anzuklicken. Als letztes müssen Sie dann auch angeben, wie Sie bezahlen möchten. Neben verschiedenen Kreditarten können Sie die Gebühren auch von Ihrem Konto abbuchen lassen. Über den *Proceed*-Schalter gelangen Sie zum nächsten Fenster, in dem Sie dann Angaben tätigen müssen wie beispielsweise Ihren Namen, Telefonnummer, Anschrift und Kreditkartennummer.

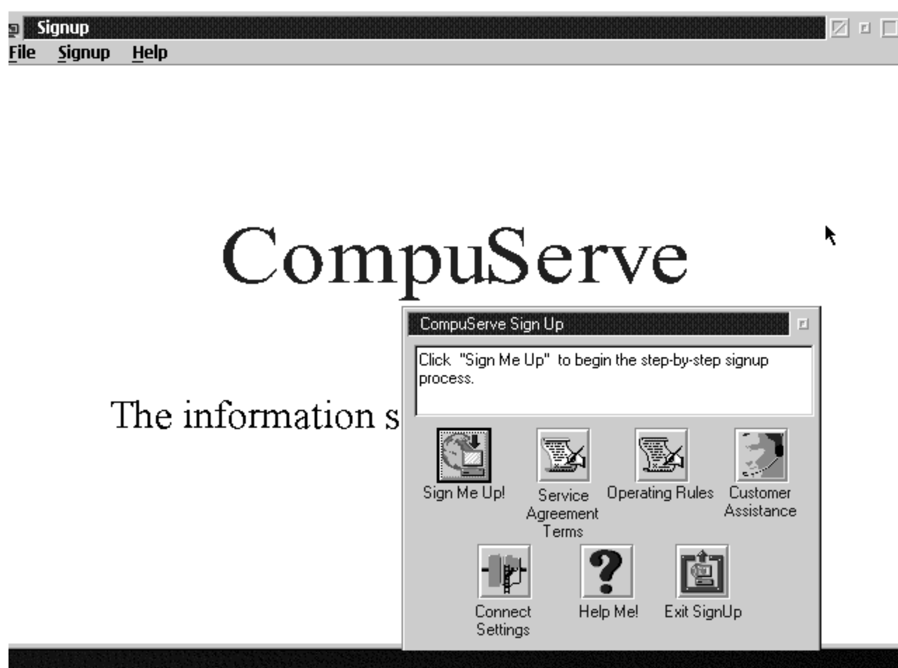


Abb. 11.22: »Schritt für Schritt« ist CIM zu konfigurieren

Im Fenster *Phone Liste* wählen Sie aus einer Liste einen Anwahlpunkt zum CompuServe aus. Er sollte möglichst in Ihrer Stadt oder halbwegs in der unmittelbaren Umgebung liegen, so daß die Telefonkosten noch vertretbar sind. Beachten Sie auch, daß es für jeden Anwahlpunkt auch verschiedene Baudraten gibt. Zusätzlich müssen Sie dann auch im nächsten Fenster das Wahlverfahren Ton oder Impuls sowie die genutzte Schnittstelle (Com1-Com4) vorgeben. Wissen Sie nicht, an welcher Schnittstelle das Modem angeschlossen ist, können Sie auch Auto-Detect anklicken. Nachdem Sie diese Eingaben getätigt haben, wird die Online-Registrierung durch-

geführt. Haben Sie dann der Einverständniserklärung von CompuServe zugestimmt, wird Ihnen in einem Fenster Ihre User ID und das Paßwort mitgeteilt. Die User ID ist Ihre Mitgliedsnummer, über die Sie auch Emails empfangen können. Notieren Sie sich die beiden Codes, denn im nächsten Fenster müssen Sie diese zur Kontrolle nochmals eintippen.

11.3.2 Konfiguration über den CompuServe Information Manager

Haben Sie bereits einen CompuServe-Account, brauchen Sie die vorher beschriebene Prozedur nicht durchlaufen. Sie starten direkt aus dem Ordner CompuServe das Programm CIM for OS/2.

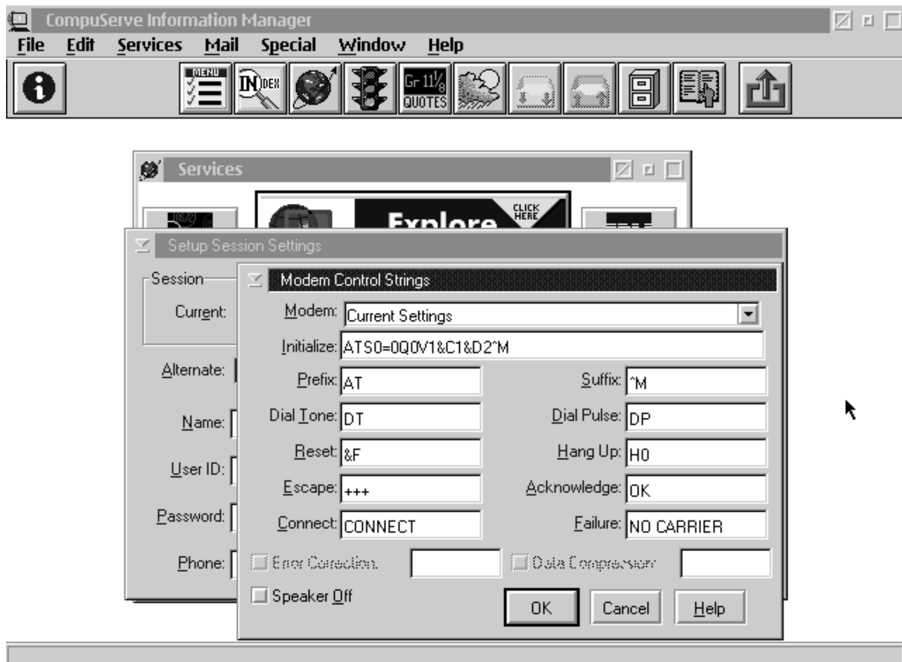


Abb. 11.23: Im Setup-Settings-Fenster können Sie die Einstellungen ändern

Öffnen Sie das Special-Menü und wählen Sie *Session Settings* aus. Es öffnet sich ein Fenster mit der gleichnamigen Bezeichnung. Session Settings benötigen Sie nicht nur, wenn Sie den CompuServe Information Manager zum ersten Mal konfigurieren wollen, sondern auch, wenn Sie Änderungen an der Hardware, User ID oder am Wahlverfahren vornehmen möchten.

Tragen Sie nun nacheinander Ihren Namen, die User ID, die benutzte Schnittstelle COM1-COM4 sowie die Baudrate Ihres Modems ein. Auch ist das Wahlverfahren Ton oder Puls zu bestimmen. Klicken Sie Modem an und wählen Sie aus der Liste den Modemtyp aus. Sollte er

nicht in der Liste aufgeführt sein, sollten Sie *Hayes* aktivieren. Die Modemkommandos werden entsprechend des ausgewählten Modemtyps in die Felder eingetragen. In der Regel sind an diesen Eintragungen keine Änderungen vorzunehmen. Sie können dann nach der Bestätigung der Angaben sofort eine Verbindung aufbauen.

Weitere Detaileinstellungen, beispielsweise in welches Verzeichnis heruntergeladene Software gespeichert werden oder wie die Anzeige im Telefonbuch erfolgen soll, machen Sie wieder im Special-Menü. Wählen Sie dann die Option *Preferences* aus.

Als nächstes sollten Sie die Druckereinstellungen kontrollieren. Das ist insbesondere wichtig, wenn Sie auch ein Faxprogramm installiert haben, das über die Druckoption jeglicher OS/2-Anwendungen aktiviert werden kann. Über das File-Menü sowie *Print Settings* markieren Sie den gewünschte Drucker. Mit *Properties* können Sie dann auch weitere Einstellungen wie das Format und die Auflösung vorgeben.

11.3.3 In CompuServe navigieren

Nachdem Sie das Programm erfolgreich konfiguriert haben, können Sie nun als nächstes eine Verbindung aufbauen. Dafür öffnen Sie das File-Menü, um die Option*Connect* auszuwählen. Wird der CompuServe Information Manager erst neu aktiviert, öffnet sich automatisch ein Fenster, worüber der Verbindungsaufbau erfolgen kann. Wahlweise können Sie den Verbindungsaufbau auch anders realisieren, klicken Sie einfach ein Symbol beispielsweise aus dem Services-Fenster an. Automatisch wird die Verbindung aufgebaut, und auf Ihrem Bildschirm sehen Sie den Inhalt des ausgewählten Services.

Sie haben den Zugang zu CompuServe und sehen auf Ihrem Bildschirm das Fenster *Services*. Nach Themen sortiert verbergen sich hinter den Symbolen Nachrichten, Neuigkeiten, Foren und Software sowie Einkaufmöglichkeiten. Öffnen Sie ein Symbol mit einen Doppelklick. Wahlweise ist das auch über das Services-Menü realisierbar.



Abb. 11.24: Gezielt können Sie nach Themen suchen

Da in der Regel gezielt nach Informationen gesucht wird, gibt es dafür eine Find-Funktion. Wahlweise ist diese Funktion über das Services-Menü oder über das Indexsymbol aus der Symbolleiste zu aktivieren. Geben Sie den Suchbegriff ein. In einem Fenster werden Ihnen alle gefundenen Einträge angezeigt. Möchten Sie einen Eintrag in der Liste anschauen, brauchen Sie ihn nur zu markieren und den Go-Schalter anklicken.

Ist Ihnen die Bezeichnung eines gesuchten Services bekannt, können Sie diese auch direkt eingeben. Die manchmal etwas zeitaufwendige Suche über die Find-Funktion entfällt somit.

Zusätzlich ist am Bildschirmrand eine Symbolleiste. Darüber können Sie sich den Inhalt der aktivierten Sektion anzeigen lassen, gezielt darin suchen, durch die Eingabe eines Suchkriteriums Dateien laden und auch sichern.



Abb. 11.25: Ihre Lieblingsforen

Möchten Sie öfters an speziellen Foren teilnehmen oder immer die aktuellsten Spiegelnachrichten lesen, dann sollten Sie die Option *Favorite Place* nutzen. Darüber können Sie jegliche Einträge sehr schnell wiederfinden und jederzeit aktivieren. Sie interessieren sich beispielsweise für deutschen OS/2-Anwendungen. In regelmäßigen Abständen schauen Sie im IBM-OS/2-Forum nach, ob neue interessante Programme in der Rubrik »Dt. OS/2-Anwendungen« vorgestellt werden. Das Forum tragen Sie über das Menüsymbol oder über die Option *Favorite Places* und *Add* in ihre Liste ein.

Möchten Sie einen Services aus der Liste aktivieren, brauchen Sie nur das Menüsymbol anklicken sowie den gewünschten Eintrag auswählen. Automatisch wird die Verbindung zu CompuServe hergestellt und der Service geladen.

11.3.4 Teilnahme an Foren

Wie eingangs schon erwähnt, bietet CompuServe die Möglichkeit, an einer Vielzahl von Foren aktiv teilzunehmen. Dabei ist die Themenvielfalt nahezu unüberschaubar. Bleiben wir bei unserem IBM-OS/2-Forum (OS2UGER). Über GO oder dem Index-Symbol haben Sie die Verbindung zu CompuServe als auch dem Forum aufgebaut. Öffnen Sie beispielsweise die »Plauderecke«.

Um sich die Beiträge anzuschauen und vielleicht auch gleich selbst darauf zu antworten, müssen Sie den Beitrag mit der Maus anklicken oder über den Get-Schalter öffnen. Nun können Sie den Beitrag lesen. Möchten Sie auch Stellung dazu nehmen, klicken Sie den Replay-Schalter an. Sofort öffnet sich die Mail-Funktion, wo bereits die Empfängerangaben eingetragen sind. Schreiben Sie nun Ihren Kommentar, und wahlweise versenden Sie ihn gleich oder legen ihn in den Postausgangskorb ab.

Sie können aber auch mit Teilnehmern eines Forums den direkten Kontakt aufnehmen. Aktivieren Sie nun aus der beschriebenen Symbolleiste das Gesichtsymbol. In einer Liste werden die Namen von den Teilnehmern aufgeführt, die gerade irgendwo auf der Welt am Forum teilnehmen. Ihr Name muß dort auch aufgeführt sein. Sie können nun direkt zu den Teilnehmern den Kontakt aufbauen und sich auf diese direkte Art zu einem Themen austauschen.

11.3.5 Shopping

Wie auch beim Einkaufen per Btx, sind Sie bei CompuServe an keine Geschäftszeiten gebunden. Vielfältig ist das Angebot und oft auch eine Versuchung. Um einen Einkaufsbummel zu unternehmen, müssen Sie aus dem Services-Fenster das Shopping-Symbol anklicken.

Wählen Sie aus der Electronic Mail/Shopping-Liste den gewünschten Anbieter aus. Es wird Ihnen dann entweder eine Liste der unterschiedlichen Produktgruppen angezeigt oder gleich eine Artikelliste. Mit der Maus markieren Sie den gewünschten Artikel und geben auch die Anzahl an. Über den Order-Schalter ist die Bestellung vollzogen.

11.3.6 Software laden

Auch über CompuServe können Sie Programme auf Ihren PC übertragen. Das Spektrum reicht vom Shareware- bis hin zu kommerziellen Lösungen. Um nach bestimmten Softwareherstellern oder Softwarekategorien zu suchen, sollten Sie die Find-Funktion nutzen. Möchten Sie sich erst einmal einen Überblick verschaffen – soweit das möglich ist – was für Software über CompuServe übertragen werden kann, sollten Sie wiederum die Find-Funktion nutzen und beispielsweise »Shareware« eintippen. Über den Ok-Schalter wird automatisch die Verbindung zu CompuServe aufgebaut, und im Fenster *Search result* sehen Sie das Ergebnis der Suche.

Es öffnet sich ein Fenster, in dem Ihnen in der linken Liste alle gefundenen Einträge angezeigt werden. Sie können über den Rollbalken darin blättern und über den Schalter *Press fore more items* veranlassen, daß weitere Einträge sichtbar werden. Per Mausklick ist die entsprechende Produktgruppe oder der Hersteller auszusuchen. Über den GO-Schalter wechseln Sie zu dem Listeneintrag. Als nächstes öffnet sich ein Hinweisfenster des ausgewählten Anbieters. Klicken Sie den Join-Schalter an, um das komplette Angebot des Anbieters einsehen zu können.

Sie haben nun auf Ihrem Bildschirm das Logo des Anbieters und auch die Tool-Symbolleiste, wo Sie das Bibliothekssymbol anklicken. Es öffnet sich nun ein weiteres Fenster mit der Bezeichnung *Library Section*. Hier sind die Angebote nach Themen oder Produkten sortiert.

Mit *Select* treffen Sie Ihre Wahl.Über den Retrieve-Schalter wird das Fenster *Datei sichern unter* geöffnet. Wahlfrei geben Sie den Dateinamen, das Laufwerk und das Verzeichnis an, in dem die Datei auf Ihrem PC gesichert werden soll.

11.3.7 Komfortable Mail-Funktion

Neben der Teilnahme an Foren, dem Shopping oder dem Herunterladen von Software wird CompuServe zum größten Teil genutzt, um Emails zu verschicken und zu empfangen. Doch können Sie damit nicht nur einfache Nachrichten versenden, sondern auch Dateien. Dabei werden die Formate Text, Binary, GIF und JPEG unterstützt.

Emails verfassen und verschicken

Um mit Hilfe des CompuServe Information Manager ein Email zu verfassen und zu verschicken, ist aus dem Hauptmenü *Mail* und dann *Create Mail* auszuwählen. Im sich öfnnenden Fenster *Recipient List* tragen Sie den Namen des Empfängers ein. Zusätzlich finden Sie die Schalter TO, CC und BC. Über CC erreichen Sie, daß ein oder mehrere Empfänger eine Kopie der Nachricht erhalten. Der Hauptempfänger findet in dem Email einen entsprechenden Hinweis, an wen eine Kopie verschickt wurde. Soll an einen oder mehrere Empfänger eine Kopie einer Nachricht geschickt werden, ohne daß es für die anderen Adressaten erkennbar sein darf, ist die Adresse in die Zeile Bcc (Blind Carbon Copy) einzutragen.

Tragen Sie dann die Adresse (User ID) des Empfängers ein. Über den Add-Schalter werden die Angaben automatisch in die Empfängerliste hinzugefügt. So ist es auch möglich, eine Nachricht gleich an mehrere Empfänger zu schicken. Nacheinander geben Sie immer den Namen und die Adresse ein.

Hinweis: Soll die Nachricht an einen Internet-Teilnehmer verschickt werden, müssen Sie im Adreß-Feld die Internet-Adresse angeben. Vor dieser Adresse muß unbedingt Internet: stehen.

Beispiel: Die Internetadresse lautet: inah@iherbert.do.cul.de.

So muß im Adreß-Feld stehen:

Internet:inah@iherbert.do.cul.de

Im unteren Bereich des Fensters *Recipient List* finden Sie zwei Listen. In der linken Liste sehen Sie die Einträge aus dem Adreßbuch. Sollte der Empfänger bereits in dem Adreßbuch gespeichert sein, können Sie den Eintrag anklicken, und über *Copy* >> wird er in die Liste der Empfänger übernommen. Soll ein Eintrag aus der Empfängerliste in das Adreßbuch übernommen werden, ist wiederum der Copy<<-Schalter anzuklicken.

Sie können aber auch ganz gezielt nach einem Empfänger suchen, der nicht in ihrem lokalen Adreßbuch gespeichert ist, um diesen dann in die Empfängerliste einzufügen. Sind Ihnen also die Empfängerangaben nicht vollständig bekannt, können Sie über die Search-Funktion eine Verbindung zu CompuServe aufbauen und nach dem Empfängerdaten suchen. Dafür geben Sie den Suchbegriff wie den Namen ein. Diese Funktion ist direkt aus dem Fenster *Recipient Liste* zu aktivieren.

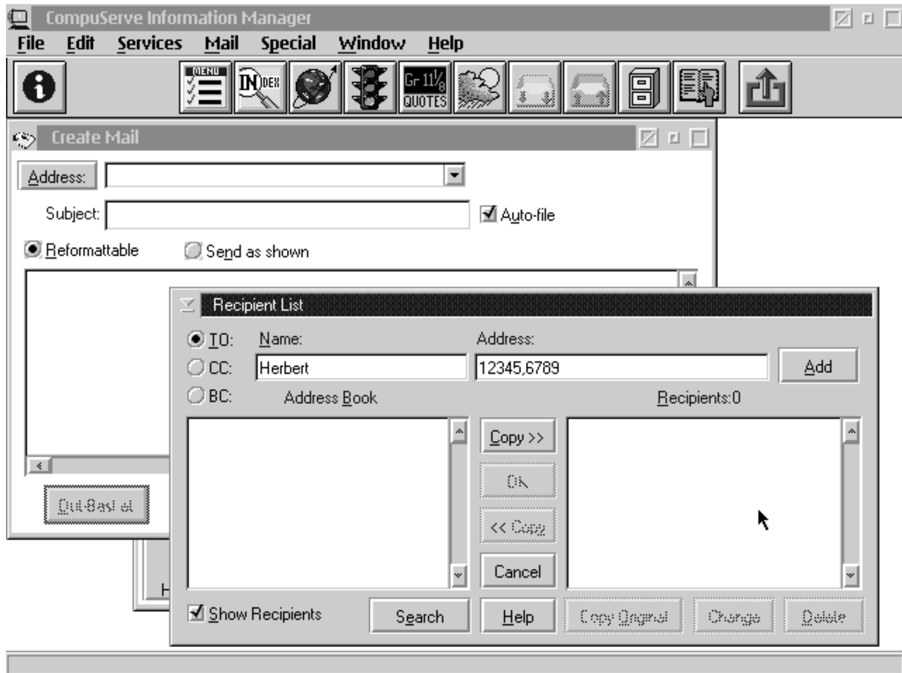


Abb. 11.26: Emails erstellen und versenden

Nachdem Sie die Empfängerangaben ermittelt haben und sie auch in der Empfängerliste eingetragen sind, sind die Angaben über den Ok-Schalter zu bestätigen. Im oberen Bereich des Fensters *Create Mail* wird Ihnen die Anzahl der Empfänger Ihrer Nachricht eingeblendet. Möchten Sie an den Adreßangaben Korrekturen vornehmen, wechseln Sie über den Adreß-Schalter nochmals zum vorherigen Fenster.

Tragen Sie nun in der Zeile *Subject* den Betreff-Text ein. Er sollte kurz und aussagekräftig sein. Sie können ein Email nur verschicken, wenn Sie in der Betreff-Zeile einen Eintrag vornehmen. Schreiben Sie nun in den Eingabebereich die Nachricht an den Empfänger.

Für weitere Detaileinstellungen klicken sie dann *Options* an. Hier können Sie beispielsweise eingeben, welche Priorität (Normal, Low, High) die Nachricht hat. Auch ist zu bestimmen, ob der Sender, der Empfänger oder beide für die Verbindungsgebühren aufkommen.

Dateien versenden

Im Mail-Menü versenden Sie über die Option *Send File* Dateien. Wie auch beim Emailversand ist der Name und die User ID manuell einzutippen oder aus dem lokalem Adreßbuch auszuwählen. Auch steht die Option zur Verfügung, eine Verbindung zu CompuServe aufzubauen, um dann nach den Empfängerangaben zu suchen.

Hinweis: Soll die Datei an einen Internet-Teilnehmer verschickt werden, muß auch hier wieder im Adress-Feld die Internet-Adresse angegeben werden.

Für die Dateiübertragung in »fremde« Netze fallen zusätzlich Kosten an. Nur wenn Sie bestätigen, daß Sie dafür aufkommen, erfolgt der Transfer.

Nachdem Sie die Empfängerangaben eingetragen, beziehungsweise ermittelt haben, ist über den Ok-Schalter zum Fenster Send File Message zu wechseln. Geben Sie nun in der Zeile *Subject* den Betreff-Text ein. Über *Options* ist dann wiederum anzugeben, wer für die Gebühren aufkommt, sowie welche Priorität die Sendung hat.

Klicken Sie dann den File-Schalter an, um den Dateinamen, das Verzeichnis und Laufwerk anzugeben, in dem die zu sendende Datei gespeichert ist. Diese Angaben werden in die File-Zeile übernommen. Es ist leider nur möglich, jeweils eine Datei zu verschicken. Zusätzlich müssen Sie das Format der Datei angeben. Zwischen Text, GIF, Binary und JPEG ist zu wählen.

Emails abholen

Um herauszufinden, ob Emails für Sie abzuholen sind, müssen Sie zu CompuServe eine Verbindung aufbauen. Dafür ist aus dem File-Menü *Connect* auszuwählen. Ist Post für Sie da, wird das durch ein Briefsymbol in der Symbolleiste des Programms automatisch angezeigt. Damit Sie die Liste der eingegangenen Mails einsehen können, ist das Briefsymbol anzuklicken.

Markieren Sie in der Liste die Mails, die Sie einsehen möchten, und klicken Sie den Get-Schalter an. Wahlweise können Nachrichten in den Posteingangskorb gelegt oder über *File It* für den späteren Gebrauch gespeichert werden. Natürlich ist eine Nachricht auch gleich zu löschen. Wollen Sie weitere Empfänger von der Nachricht informieren, dann ist der Forward-Schalter anzuklicken. Es öffnet sich dann das bekannte Fenster zum Verschicken einer Nachricht.

11.3.8 Kassensturz

Damit die nächste Kreditkartenabrechnung nicht zur großen Überraschung wird, können Sie sich jederzeit über Ihren aktuellen Kontostand bei CompuServe informieren.

Stellen Sie über das File-Menü und Connect ein Verbindung zu CompuServe her. Klicken Sie dann im Fenster *Services* das Symbol *Member Services* an. Hier können Sie nicht nur Ihren aktuellen Kontostand abfragen, sondern auch andere Informationen wie Telefonnummern ermitteln.

Um Ihrem Kontostand abzufragen, ist *Account/Billing Administration* sowie der Select-Schalter anzuklicken. Im nächsten Fenster müssen Sie dann wiederum *Direct Debit Status* und dann *Review Your Charges* aktivieren. Es öffnet sich ein Fenster mit der Bezeichnung *Terminal Emulation*, in dem das System zur zeichenorientierten Darstellung umschaltet. Über ein Zahlenmenü können Sie nun zwischen den verschiedensten Option wählen können.

Unbedingt sollten Sie diesen Service in die Liste *Favorite Places* aufnehmen.

11.4 FaxWorks für OS/2

von Ina Herbert

Mit FaxWorks für OS/2 können Sie direkt aus dem PC Faxe verschicken und auch empfangen. FaxWorks ermöglicht Ihnen aus jeder Applikation heraus oder per Drag&Drop Dateien an FaxWorks zu übergeben. Sie können eigene Faxdeckblätter erstellen oder auch das Standardfaxdeckblatt des Programms nutzen. Wahlweise kann jede Seite des Faxes mit einer Anmerkung und auch mit Kopfzeilen versehen werden.

Damit nicht immer die Faxnummer manuell eingetippt werden muß, kann mit FaxWorks auch ein Telefonbuch eingerichtet werden. Per Mausklick werden die Empfängerdaten in das aktuelle Fax übernommen. Über das Telefonbuch ist es auch möglich, eine Empfängergruppe zu bilden, an die ein identisches Faxschreiben verschickt werden soll.

Wahlweise können Sie ein Fax sofort verschicken oder über die Zeitversetzt-Funktion die Uhrzeit sowie das Datum angeben. So ist es möglich, den »Mondschein«-Tarif der Telekom zu nutzen, ohne daß dafür der PC beaufsichtigt werden muß. Der Versand erfolgt automatisch anhand der Datums- und Zeitvorgabe.

Eine weitere interessante Funktion des Faxprogramms ist die Möglichkeit, daß Sie erst mit einem anderen Teilnehmer telefonieren und dann sofort über die gleiche Verbindung ein Fax schicken können.

11.4.1 Konfiguration des Programms

Bevor Sie überhaupt ein Fax verschicken können, muß das Programm konfiguriert werden. Beim erstmaligen Aktivieren von FaxWorks werden Sie darauf aufmerksam gemacht. Jegliche Vorgaben, auch nachträgliche Änderungen an dem System, werden über das Settings-Menü vorgenommen.

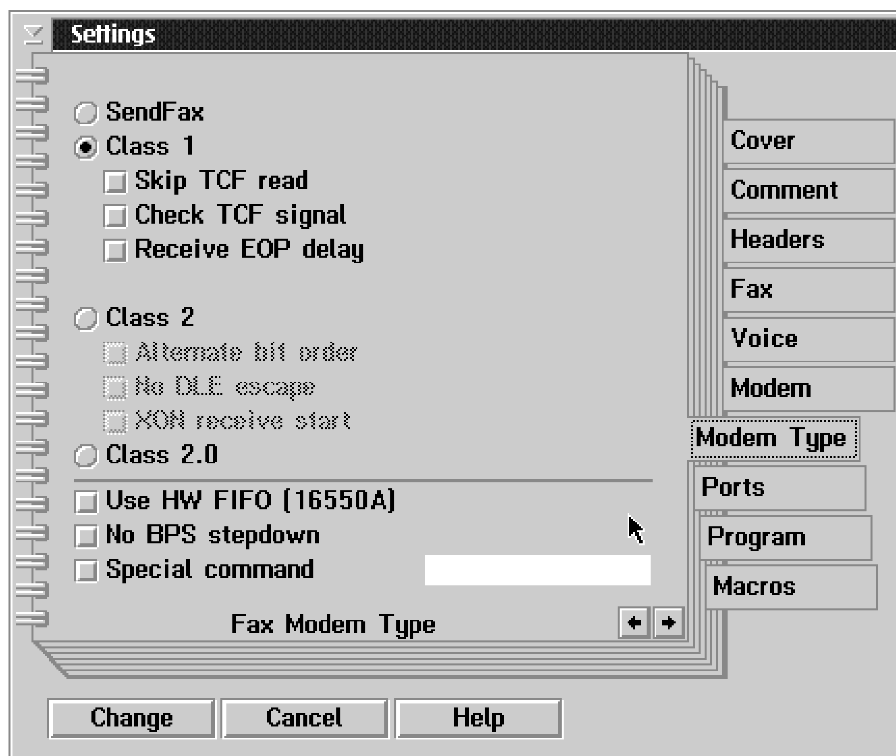


Abb. 11.27. Über das Settings-Menü ist das Programm zu konfigurieren

Nachdem Sie FaxWorks aktiviert haben, öffnet sich das Settings-Menü. Über die Registerzungen am rechten Fensterrand finden Sie die Untergliederung des Menüs. Diese sind nun nacheinander »aufzuschlagen«.

Beginnen wir mit dem Deckblatt/Cover. Hier können Sie bestimmen, ob ein Faxdeckblatt mit verschickt werden und es vom Umfang her eine ganze Seite einnehmen soll. Zusätzlich bietet FaxWorks die Möglichkeit, auf dem Faxdeckblatt eine Grafikdatei mit auszugeben. So könnten Sie beispielsweise Ihr Firmenlogo auf dem Deckblatt integrieren. Die Grafikdatei muß im Bitmap-Format vorliegen. Als nächstes sind nun Ihre Absenderdaten einzutragen.

Nachdem die Vorgaben für das Deckblatt abgeschlossen sind, ist per Mausklick das nächste Register aufzuschlagen. Hier müssen Sie für das Deckblatt die Schriftart angeben. Sie können auch einen Anmerkungstext eintragen, der dann auf allen Fax-Mitteilungen ausgegeben wird.

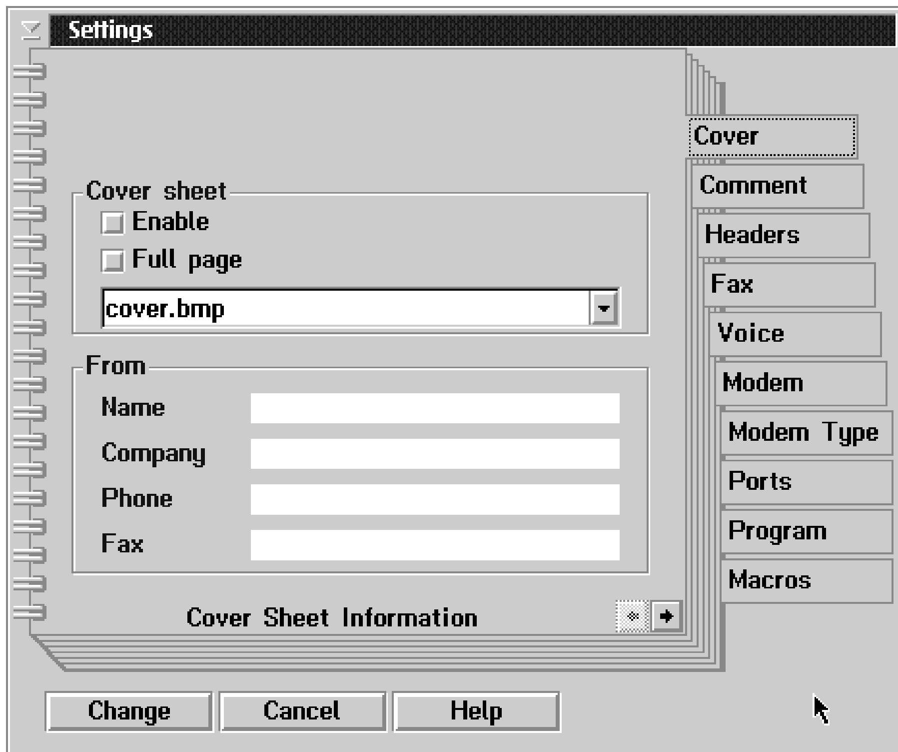


Abb. 11.28: Für das Faxdeckblatt sind verschiedene Einstellungen vorzunehmen

Damit auf jeder Seite des Faxes eine Kopfzeile erscheint, ist in dem nächsten Register *Kopfzeile/Header* der entsprechende Text einzugeben. Empfehlenswert ist beispielsweise, Ihren Namen mit der Telefon- und Faxnummer anzugeben. So kann der Empfänger eingehende Faxe besser zuordnen. Sollen in der Kopfzeile auch die Empfängerdaten ausgeben werden, muß das Feld *An* aktiviert sein.

Im Register Fax können Sie nun Ihre Fax-Kennung angeben. Wird eine Telefonanlage eingesetzt, muß in der Regel eine Nummer vorgewählt werden, um eine Amtsleitung zu erhalten. Diese Nummer sollte, gefolgt von einigen Kommata, in das Feld *Vorwahl wählen* eingegeben werden. Durch die Kommata erreichen Sie eine Verzögerung des Wahlvorgangs, und FaxWorks wartet auf die Amtsleitung. Jedes Komma verschafft eine Pause von ca. 2 Sekunden.

Zusätzlich ist dem System mitzuteilen, wie oft es versuchen soll, eine Gegenstelle zu erreichen, wenn diese besetzt ist. Neben der Anzahl der Übermittlungsversuche können Sie auch die Wartezeit in Minuten zwischen den Versuchen vorgeben. Dabei verursacht jeder Sendeversuch einen Eintrag in das Fax-Logbuch.

Wollen Sie FaxWorks nicht nur zum Faxen, sondern auch als Anrufbeantworter nutzen, dann müssen Sie Ihre Vorgaben, z.B. welche Antwortdatei bei einem Anruf abgespielt werden soll, im Voice-Register machen. Da es sich allerdings bei der FaxWorks-Version um eine eingeschränkte Version handelt, ist die Voice-Funktion auch nicht vollwertig.

Um dann als nächstes das Modem zu konfigurieren, die Übertragungsrate und das Wählverfahren einzugeben sowie wahlweise den Lautsprecher ein- oder abzuschalten, ist in das nächste Register zu verzweigen. Auch können Sie hier die Anzahl der Ruftöne einstellen, bevor Ihr Fax/Modem abnimmt. Die standardmäßig vorgegebene Einstellung von ein Mal können Sie getrost auf drei hochsetzen.

Über das Register *Modem-Typ* können Sie die von FaxWorks automatisch vorgenommenen Einstellungen über den erforderlichen Befehlssatz ändern. Diese Einstellungen basieren auf Ihren Angaben, die Sie bei der Installation des Programms getätigt haben. So geben Sie beispielsweise an, ob Ihr Fax/Modem besondere Initialisierungsbefehle benötigt oder Ihr Modem über die Hochleistungspufferung im 16550A-Chip verfügt. Detaillierte Angaben finden Sie dazu in der Dokumentation Ihres Modems.

Nachdem Sie den Fax/Modem-Typ bestimmt haben, können Sie ihn über das Register *Anschlüsse* ändern beziehungsweise anpassen. In einer Liste werden die Faxanschlüsse Ihres Systems angezeigt. Möchten Sie den Status eines Anschlusses modifizieren, beispielsweise von Senden/Empfangen auf nur Senden, ist der Eintrag anzuklicken. Der Status kann auf *Senden/Empfangen*, *nur Senden*, *nur Empfangen*, *Standby* und *Aus* geändert werden.

Im nachfolgenden Register *Programm* ist dann der eingesetzte Monitortyp anzugeben. Auch können Sie festlegen, ob Sie akustisch über den erfolgreichen Empfang oder Versand eines Faxes informiert werden wollen.

Im letzten Register haben Sie die Möglichkeit, bereits vordefinierte Makros mit einzubinden. Sie sind standardmäßig bereits in die Eingabezeilen eingetragen. Diese Makros sind sogenannte Wählmakros, d.h., sie erlauben, anstatt komplizierter Anwahlsequenzen einfachere Nummerncodes zu verwenden. So dient der Makro LCL für die Amtsholung; LD steht für Ferngespräche, INTL für Auslandgespräche und CC für Telefonkartennummern.

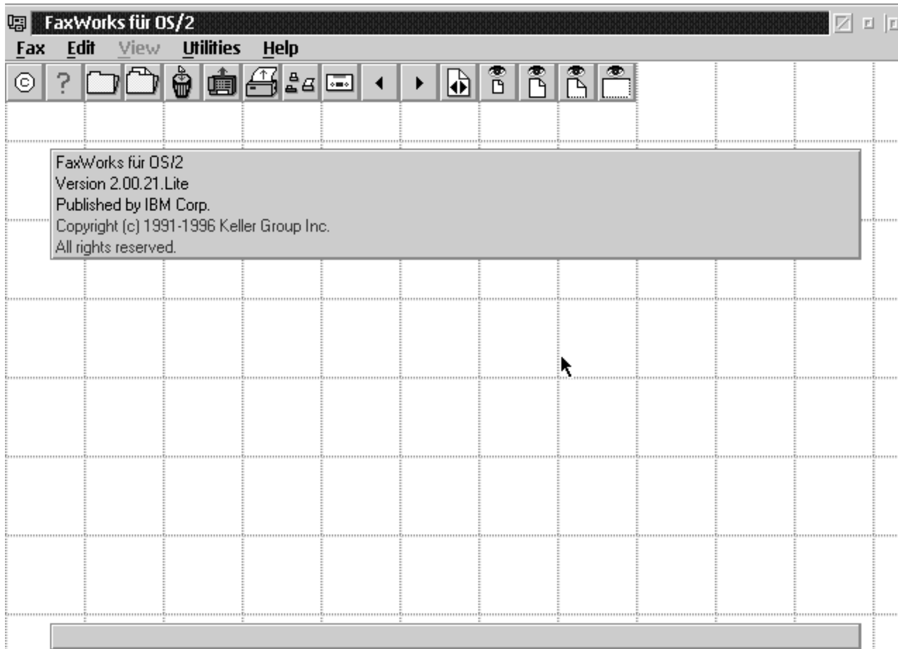


Abb. 11.29: Die Benutzeroberfläche von FaxWorks Lite

Als nächstes ist nun noch die Druckereinstellung zu überprüfen. Dafür wählen Sie aus dem Hauptmenü von FaxWorks die Option *Fax* und dann *Druckereinstellungen/Printer-Setup* aus. In einer Liste werden Ihnen alle installierten Drucker sowie FxPrint angezeigt. Diesen Eintrag müssen Sie markieren, um dann den Schalter Jobmerkmale/Job properties anzuklicken. Es öffnet sich ein Fenster mit der Bezeichnung »FxPrint Lite 32bit Printer Driver Configuration«.

Bestimmen Sie nun die Seitenlänge des Faxes. Dabei können Sie beispielsweise zwischen den Formaten Letter, A4 oder Legal und Andere wählen. Wünschen Sie ein anderes Format, ist die individuelle Seitenlänge in cm einzutragen. Zusätzlich geben Sie an, ob das Quer- oder Hochformat gewünscht wird sowie Auflösung und Graustufen. Bei der feinen Auflösung wird das Fax bei 200x200 dpi und bei der normalen Auflösung mit 200x100 dpi ausgegeben.

Auch bestimmen Sie, ob Sie ein sofortiges Senden von Faxmitteilungen wünschen oder ein späteres Senden mit Hilfe des Faxprogramms FaxWorks. Soll die Datei erst ins Logbuch eingetragen werden, dann können Sie das Fax mit Hilfe von FaxWorks einsehen, editieren und natürlich jederzeit versenden.

Zuletzt wählen Sie dann die Druckertypen für die LPT-Emulation aus. Diese Auswahl hat keinen Einfluß auf das Drucken von anderen OS/2-Anwendungen, in denen Sie eigene Druckertreiber ausgewählt haben.

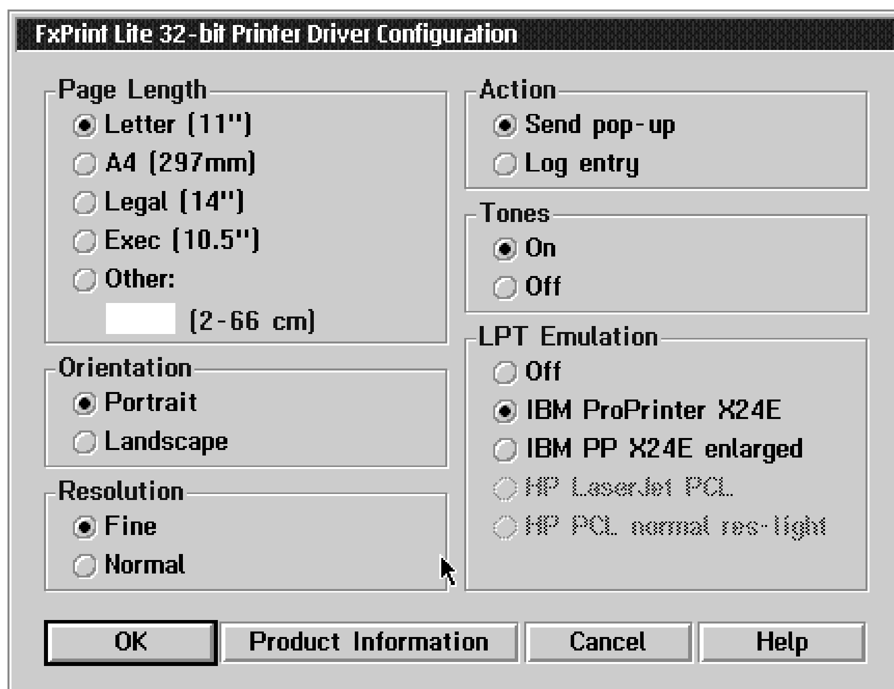


Abb. 11.30 Die Druckertreiberkonfiguration für FaxWorks

Nachdem Sie alle Angaben auf ihre Richtigkeit überprüft haben, ist der Ok-Schalter anzuklicken. Durch die Konfiguration des FxPrint-Druckertreibers können Sie nun aus jeder beliebigen Anwendung heraus Dateien als Faxe verschicken. Wählen Sie dann in dem genutzten Anwendungsprogramm das Drucker-einrichten-Menü aus, um den FxPrint-Druckertreiber auszuwählen.

11.4.2 Möglichkeiten, eine Faxmitteilung zu verschicken

Sie haben mit FaxWorks drei Möglichkeiten, Faxe zu verschicken. Entweder Sie verfassen und senden das Fax direkt mit dem Faxprogramm oder auch aus einer anderen Anwendung heraus. Wird das Fax mit FaxWorks erstellt, können Sie den Text lediglich auf das Faxdeckblatt schreiben. Der Platz dafür ist großzügig bemessen. Wollen Sie mehrere Seiten Text und/oder auch Grafiken versenden, müssen Sie die Faxoption aus dem Anwendungsprogramm heraus aktivieren. Eine weitere Variante zum Verschicken einer Datei ist es, diese per Drag&Drop auf das FxPrint-Symbol zu ziehen.

Fax direkt mit FaxWorks versenden

Sie haben FaxWorks aktiviert und öffnen das Fax-Menü, wo Sie wiederum *Senden* auswählen. Soweit Sie Ihre Absenderdaten im Einstellungen-Menü bereits eingetragen haben, wurden diese in das Fenster übernommen. Als nächstes geben Sie die Empfängerangaben ein. Möchten Sie an diesen Empfänger öfters ein Fax schicken, sollten Sie den Schalter *Telefonbuch* anklicken, damit die Angaben darin gespeichert werden. Erhält der Empfänger voraussichtlich nur einmal ein Fax, sind die Angaben manuell einzutippen. Auch können Sie über das Telefonbuch einen bereits erfaßten Teilnehmer auswählen. Die Daten werden automatisch in den Bereich *An:* übernommen.

In den Bereich *Anmerkungen* schreiben Sie den Mitteilungstext. Sie können darin ohne Probleme einen Text von über 35 Zeilen unterbringen. Über die Vorschauoption sollten Sie das Fax vor dem Versand kontrollieren. Zusätzlich können Sie über den Schalter Mehr/More den Inhalt der Kopfzeile betrachten. Damit Sie das Fax im Logbuch besser identifizieren können, ist wahlfrei eine Notiz einzugeben.

Wahlweise können Sie das Fax sofort verschicken. Dafür klicken Sie den Schalter *Aktuelles senden/Send current* an. Mit *Zeitversetzt/Delay* senden Sie das Fax an einem anderen Tag oder zu einem anderen Zeitpunkt. Tragen Sie nur eine Uhrzeit und kein Datum ein, wird das Fax verschickt, wenn nach der Systemuhr Ihres PCs der Zeitpunkt erreicht ist. Damit der Faxversand auch nachts möglich ist, muß der PC eingeschaltet und das Faxprogramm aktiviert sein. Ist der Rechner ausgeschaltet und können vorgemerkte Faxe nicht verschicken werden, wird der Versand beim nächsten Aktivieren von FaxWorks durchgeführt.

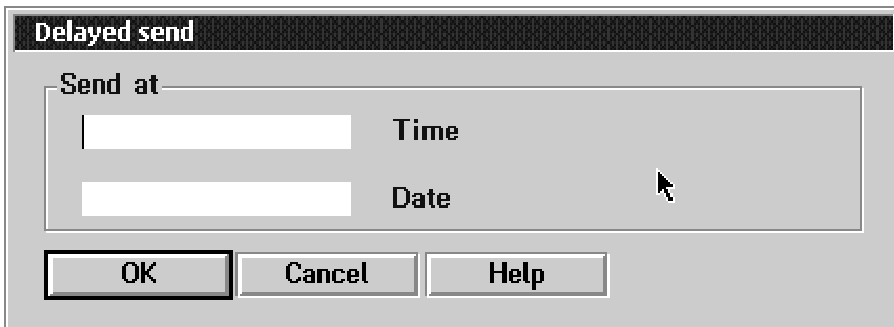


Abb. 11.31: Sowohl das Datum als auch die Uhrzeit ist vorzugeben, an dem ein Fax verschickt werden soll

Fax aus einer OS/2-Anwendung verschicken

Ohne Probleme können Sie aus einer laufenden OS/2-Anwendung ein Fax verschicken. Dafür ist es nicht notwendig, daß das Faxprogramm aktiviert ist.

Sie haben mit einem OS/2-Anwendungsprogramm eine Grafik oder Tabelle erstellt, die Sie per Fax verschicken möchten.

Als erstes sollten Sie die Datei speichern. Danach öffnen Sie das Dateimenü, in der sich in der Regel auch die Druckoption befindet. Diese müssen Sie aktivieren. Danach wählen Sie den Druckertreiber FxPrint aus, denn die Druckerausgabe soll darüber erfolgen.

Zusätzlich können Sie auch angeben, ob das komplette Dokument ausgegeben werden soll oder auch nur bestimmte Seiten. Danach ist der Druckvorgang zu starten.

Automatisch wird aus dem Anwendungsprogramm heraus FaxWorks aktiviert, und Sie haben auf Ihrem Bildschirm das Fenster »Fax senden (Deckblatt plus x Seiten)«. Ihre Absenderdaten sind bereits eintragen. Als nächstes sind nun die Empfängerangaben vorzugeben. Wahlweise können Sie bestimmen, ob ein Eintrag in das Telefonbuch erfolgen soll.

Schicken Sie dem Empfänger voraussichtlich nur einmal dieses Fax, sind die Angaben manuell einzutippen. Über das Telefonbuch kann ein bereits erfaßter Teilnehmer ausgewählt werden. Die Daten werden dann automatisch übernommen. Die weiteren Angaben wie die Anzahl der Seiten, das aktuelle Datum sowie die Uhrzeit werden automatisch in das Deckblatt eingetragen.

In dem Bereich *Anmerkungen* schreiben Sie den Mitteilungstext, also das Anschreiben an den Empfänger. Über die Vorschauoption sollten Sie das Fax vor dem Versand kontrollieren. Nur über die Vorschaufunktion ist es möglich, das Faxdeckblatt einzusehen. Die Anlagen, also die Tabelle oder die Grafik, können Sie nur betrachten, wenn Sie den Schalter *Öffnen* anklicken. Direkte Änderungen in den Faxseiten sind allerdings nicht möglich. Sie können lediglich die Ansicht vergrößern oder verkleinern oder das Fax um 180 Grad drehen. Um dann das Fax mit den Anlagen zu verschicken, ist im Fax-Menü *Senden* auszuwählen.

Zusätzlich können Sie über den Schalter *Mehr* den Inhalt der Kopfzeile kontrollieren. Damit Sie später das Fax im Logbuch eindeutig identifizieren können, ist wahlfrei eine Notiz einzugeben.

Nun können Sie bestimmen, ob das komplette Fax sofort verschickt werden soll oder an einem anderen Tag oder zu einer späteren Uhrzeit. Dies geht mit *Zeitversetzt/Delay*. Beachten Sie dabei, daß der PC und das Faxprogramm zu dem vorgegebenen Zeitpunkt aktiviert sein müssen.

Über den Schalter *Aktuelles Senden/Send current* wird sowohl der sofortige als auch spätere Faxversand aktiviert. Danach können Sie das OS/2-Anwendungsprogramm, in dem Sie die Faxanlagen erstellt haben, schließen. Zum Versenden des kompletten Faxes wird es nicht benötigt.

11.4.3 Telefonbuch nutzen

Wenn Sie direkt aus FaxWorks oder einer OS/2-Anwendung Faxe verschicken, steht Ihnen das Telefonbuch des Faxprogramms zur Verfügung. Sie können das Telefonbuch um eigene Einträge ergänzen, diese editieren und löschen. Auch kann nach Einträgen gesucht werden. Vorge-

geben ist in der FaxWorks-Version ein Telefonbuch. Sie können also nicht weitere Bücher anlegen, um die Einträge zu klassifizieren.

Um das Telefonbuch zu öffnen, ist aus der Hauptmenüleiste des Faxprogramms die Option *Fax* und dann *Senden* zu aktivieren. Sie haben dann auf Ihrem Bildschirm das Fenster zum Versenden eines Faxes. Um mit dem Telefonbuch arbeiten zu können, müssen Sie den gleichnamigen Schalter anklicken.

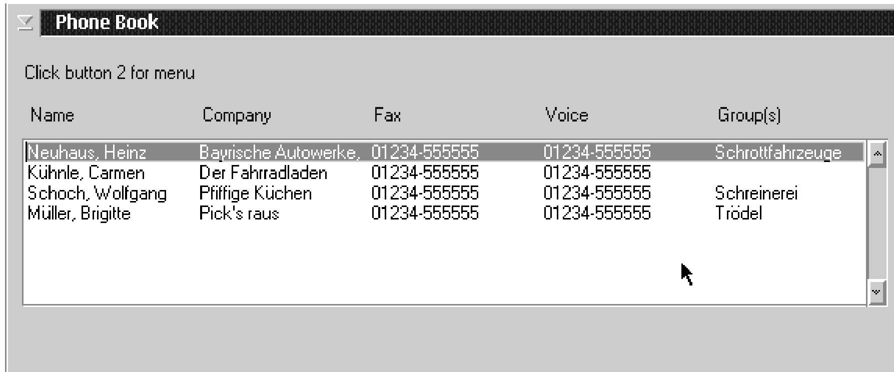


Abb. 11.32: Das Telefonbuch von FaxWorks

Wie Sie auch in der Abbildung sehen, verfügt das Telefonbuch weder über eine Menüleiste noch über Schalter, worüber die Funktionen aktiviert werden können. Möchten Sie einen Eintrag bearbeiten, müssen Sie ihn mit der Maus anklicken, so daß er markiert ist und dann die rechte Maustaste drücken. Es öffnet sich ein Fenster, in dem die Funktionen wie Editieren, Löschen, Sortieren und Auswahl bestätigen aufzurufen sind.

Möchten Sie einen Eintrag aus dem Telefonbuch ändern, muß der Datensatz markiert werden. Über die rechte Maustaste öffnen Sie das Menü, und wählen dann die Editier-Option aus. In einem weiteren Fenster können Sie alle Felder bearbeiten. Über dieses Fenster können Sie auch neue Datensätze erfassen.

Ausgesprochen dürftig ist die Suchfunktion des Programms. Sie ist wenig hilfreich, wenn größere Datenbestände in dem Telefonbuch geführt werden. Um nach einen Eintrag zu suchen, ist lediglich der Buchstabe zu drücken, womit der Name oder Vorname des gesuchten Teilnehmers beginnt. Es wird dann entsprechend der erste zutreffende Name/Vorname markiert.

Da die Datenerfassung immer eine große Fehlerquelle ist, sollten Sie sehr sorgsam die neuen Feldinhalte überprüfen, bevor sie gespeichert werden. Das Programm nimmt dabei keine Fehlerprüfungen vor. Selbst wenn Sie ins Faxnummern-Feld Buchstaben oder sonstige Zeichen eintragen, werden diese akzeptiert.

Werden mehrere Datensätze hintereinander eingetippt, ist der Neu-Schalter anzuklicken. Die Einträge werden dann nacheinander in die Liste übernommen.

Faxein- und -ausgänge kontrollieren

Jederzeit können Sie sich über das Logbuch von FaxWorks einen Überblick über die Faxaktivität sowie deren Status verschaffen. Es werden die Faxmitteilungen aufgeführt, die versendet, empfangen oder editiert wurden. Direkt aus dem Logbuch können Faxe nochmals gesendet, ausgedruckt und gelöscht werden.

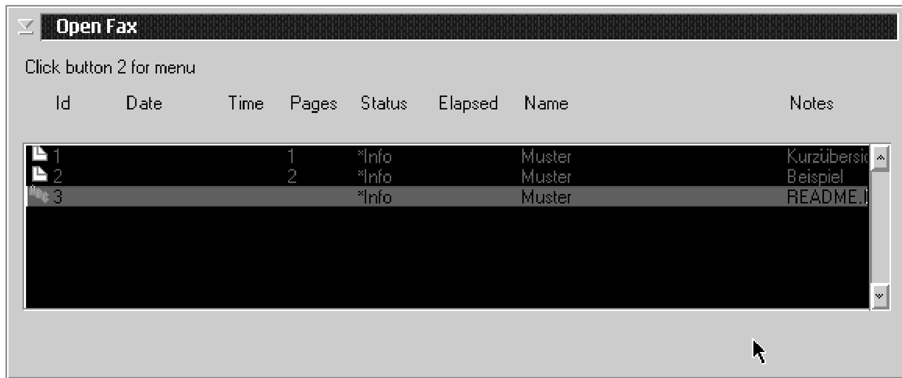


Abb. 11.33: Das Faxlogbuch von FaxWorks

Um das Logbuch zu öffnen, müssen Sie aus dem FaxWorks-Menü *Fax* und dann *Logbuch/Open Log* auswählen. Übersichtlich in tabellarischer Form können Sie sofort den Status jedes einzelnen Faxes erkennen. Neben Uhrzeit, Datum und Name wird auch der Inhalt des Notizfeldes angezeigt.

Soll ein Fax nochmals verschickt oder editiert werden, müssen Sie es mit der Maus anklicken und dann die rechte Maustaste drücken. Über ein Pop-Menü stehen Ihnen dann die Funktionen zur Auswahl. Auch können Sie so Faxe löschen, drucken oder auch den Eintrag im Logbuch bearbeiten.

Über das Popup-Menü oder die Systemsteuerung an der oberen linken Ecke schließen Sie das Logbuch wieder und gelangen zur Benutzeroberfläche von FaxWorks zurück.

11.5 VideoIn/2

von Andreas Portele

VideoIn/2 ist ein in den Bereichen Videoaufnahme (Digitalisierung) und Bearbeitung besonders leistungsfähiger Videorecorder und Videoplayer. Im Lieferumfang von VideoIN/2 befinden sich MMPM/2-Treiber zum Betrieb einer Reihe von Bildplatten und Framegrabbern, Komprimierungsroutinen für Audio und Video (CODECs) und ein Werkzeug namens »AVI-Dienstprogramm«, das zum direkten Manipulieren der AVI-Dateistruktur von Nutzen ist.



Abb. 11.34: Der VideoIn-Rekorder mit seinen Monitoren

Mit VideoIN/2 können Filme grundsätzlich in zwei verschiedenen Modi aufgenommen werden: Realtime und Framestep. Der Realtime-Modus ist zur direkten (Echtzeit-) Aufzeichnung einer »echten« Live-Videoquelle wie einem Fernseh- oder Videokamerasignal gedacht. Durch die Echtzeitverarbeitung bleibt der CPU des Rechners nur eine festgelegte kurze Zeitspanne von meistens einer 1/15 Sekunde bis einer 1/30 Sekunde, um ein Bild zu komprimieren.

Der Framestep-Modus hingegen läßt sich mehr Zeit und widmet sich der Verarbeitung von zeitunkritischen Videodaten. In diesem Modus darf die CPU pro Frame beliebig viel Zeit

beanspruchen (üblicherweise 1/4 Sekunde bis 1/2 Sekunde auf einem 486-66 bei einer Auflösung von 320x240) und liefert dafür aber auch eine höhere Komprimierungsrate und eine bessere Bildqualität. Zum Beispiel kann man in diesem Modus auf einfache Weise beliebige Formate von Videodateien ineinander überführen, indem man als Quelle die eine und als Ziel die andere Datei angibt.

Bei beiden Verfahren wird grundsätzlich auf die im MMPM/2 registrierten Kodierungsverfahren und Dateiformate zugegriffen, wodurch eine maximale Flexibilität in der Wahl der Aufzeichnungsverfahren erreicht wird. So ist man nicht auf die bei OS/2 beigelegten Formate fixiert, sondern kann sich beliebig bei Drittherstellern mit allen Video- und Audio-Formaten dieser Welt eindecken.

VideoIN/2 wurde preisgekrönt, weil es das erste zu hundert Prozent auf Software basierende Recordersystem war, das Videoquellen von 30 fps (Frames per Second) aufnehmen konnte. Selbst das »Monitor-Fenster« existiert nur als reine Softwarelösung. In diesem Monitorfenster kann man wärend der Aufzeichnung die Bildquelle überwachen und hat dabei noch die Möglichkeit, neben Helligkeit, Kontrast und Farbtemperatur außerdem noch einen Clip-Bereich festzulegen, um die bekannten schwarzen Kinobalken oder Schwächen der Bildqualität am oberen oder unteren Rand abzuschneiden. Dieser Clip-Bereich ist auch sehr praktisch, um einfach nur einen Bildauschnitt des gesammten Videobildes aufzuzeichnen. Natürlich kann VideoIN/2 auch mit hardwareunterstützen Komprimierungsalgorithmen, wie den üblichen MJPEG- (Motion JPEG) Framegrabbern arbeiten. Falls sie keinen Framegrabber in Ihrem Computer installiert haben, wird sie VideoIN/2 zunächst mit einem Nachrichtenfenster mit der Nummer »SWVR137« und einem dazu erklärenden Text begrüßen. Diese Nachricht sagt nichts anderes aus, als daß sie eben keinen Framegrabber installiert haben. Sie dürfen diese Meldung bedenkenlos mit *OK* quittieren, VideoIN/2 steht Ihnen anschließend im vollem Umfang zur Verfügung. Beim Arbeiten mit komprimierten Videos im allgemeinen sollten sie darauf achten, daß die Bildgröße auf beiden Achsen durch 8 teilbar ist; solch eine Auflösung entspricht bei so gut wie allen Komprimierungsverfahren der inneren Auflösung des Komprimierungsalgorithmus und hilft damit, Verschnitt an Speicherplatz und Rechenzeit einzusparen.

Mit dem AVI-Dienstprogramm kann man als besonderes Feature aus Einzelbildern ein Video konstruieren. Da die Einzelbild-Dateien wieder mit Funktionen des MMPM/2 gelesen werden, hängt es ausschließlich von der Anzahl der Formate ab, die Sie installiert haben, ob Sie alle ihre Bilder lesen können. Bei OS/2 Warp 4 werden eine stattliche Anzahl von Formaten wie BMP, GIF, TGA, TIF, JPEG bis hin zu PhotoCD und einige sehr spezielle und professionelle Bildformate oder sogar FAX-Formate mitgeliefert. Sie können also ohne Probleme mit OS/2-Standardmitteln einen Film aus Ihren Urlaubsfotos und den während des Urlaubs eingetroffenen Faxe erstellen. Das einzige, worauf sie hierbei achten müssen, ist, daß alle Bilder die gleiche Größe haben, denn wechselnde Bildgrößen sind innerhalb eines Videos natürlich nicht möglich. Die aus Einzelbildern erstellten AVIs werden zunächst grundsätzlich unkomprimiert erzeugt, dies kann sehr schnell zu Dateien jenseits der 100 Mbyte führen! Genügend freier Plattenplatz ist also anzuraten. Aufgrund der Arbeitsweise des MMPM/2 wird zusätzlich noch einmal der gleiche Plattenplatz benötigt, um das frisch erstellte AVI mittels *Speichern unter...* auch zu speichern. Bei knappen Platz kann man zu den Trick greifen, das als temporäre Datei erstellte AVI im Multimedia-Arbeitspfad aufzusuchen (ersichtlich in der Multimedia-

Konfiguration im Systemkonfigurationsordner), um es dort mittels Verschiebens in einen anderen Ordner und der Vergabe eines aussagekräftigen Namens in einen dauerhaften Zustand zu retten.

Das AVI-Dienstprogramm ist auch sehr nützlich um den Fehler »Ungültige Pufferlänge« beim Abspielen von einigen Windows-AVIs zu umgehen. Um diesen Fehler zu beseitigen, muß das betreffende AVI in das AVI-Dienstprogramm geladen werden. Anschließend wird es mittels eines Drucks auf den *Split*-Knopf zuerst in Audio- und Video-Daten aufgespalten, um es anschließend mittels des *Merge*-Knopfs wieder zu verschmelzen und abschließend mit *Datei sichern* abzuspeichern.

11.6 MarkVision und JetAdmin

von Bernd Rohrbach

Die Produkte JetAdmin von Hewlett-Packard und Markvision von Lexmark stellen Werkzeuge und Treiber zur Installation und Verwaltung von Druckern im Netzwerk zur Verfügung, die entweder über einen eingebauten Print-Server verfügen oder an eine externe Print-Server-Box angeschlossen sind. Die externen Print-Server-Boxen stellen in der Regel mindestens eine parallele Schnittstelle zur Verfügung, an die sich beliebige Drucker mit Centronics-Schnittstelle über ein handelsübliches PC-Druckerkabel anschließen lassen.

Da der Installationsvorgang und die Funktionsweise für beide Produkte ähnlich ist, wird an dieser Stelle nur auf die JetAdmin-Software von HP eingegangen.

Ist die Installation von OS/2 Warp 4 beendet, und alle gewünschten Drucker mitsamt den Druckertreibern installiert, kann jetzt bei jedem Druckerobjekt der Ausgabeport nachträglich geändert werden. Normalerweise kommen hierfür die Schnittstellen LPT1 bis LPT3, beziehungsweise COM1 bis COM4 in Frage. Für den Anschluß an einen HP-JetDirect Print-Server ist ein weiterer Ausgabe-Treiber vonnöten, der die Daten auf die entsprechende Netzwerkadresse leitet. Dieser Ausgabe-Treiber wurde mit der Auswahl der JetAdmin-Software bei der Installation auf die Festplatte kopiert, der jetzt nur noch installiert werden muß. Wie einfach dies geht, zeigen die nachfolgenden Schritte:

– *Einstellungen* des gewünschten Druckerobjekts öffnen

– *Ausgabeanschluß* aufblättern

– *Neuen Anschluß installieren* auswählen

– *HP Netzwerkanschluß* auswählen und bestätigen

– Eingabe des logischen Namens des Anschlusses

Die JetAdmin-Software wird nun automatisch gestartet und sucht nach allen im Netz befindlichen HP-Jet-Direct-Print-Servern (intern und extern).

Hinweis: Wird Ihr Drucker bzw. Print-Server nicht gefunden, überprüfen Sie bitte folgende Punkte: a) sind die Geräte eingeschaltet? b) ist das Netzwerk-Protokoll IEEE 802.x installiert?

– Nachdem die Software alle HP-Jet-Direct-Print-Server samt ihrer Netzwerkadressen gefunden hat und in einer Liste anzeigt, brauchen Sie nur noch die entsprechende Zeile mit der Maus zu markieren und die Taste *Ersetzen* zu drücken und anschließend mit der *OK*-Taste zu bestätigen. Schon ist der Anschluß installiert.

– Dem Drucker muß jetzt der neu installierte Anschluß zugeordnet werden: hierzu einmal auf den neuen Anschluß klicken.

– *Einstellungen* schließen

Das war's.

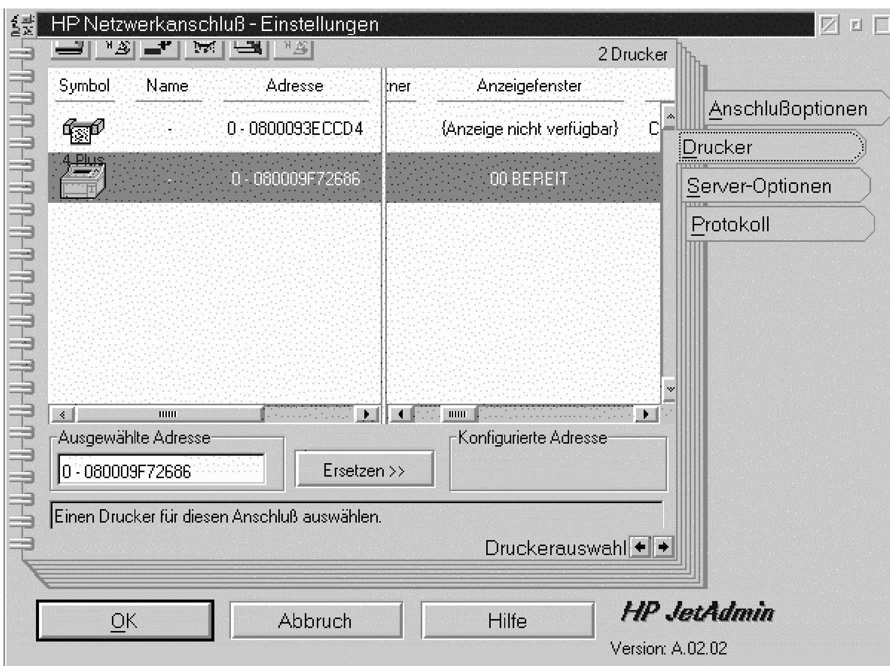


Abb. 11.35: Alle Print-Server sind gefunden worden

Hinweis: In den Einstellungen *Anschlußoptionen* des Netzwerkanschlusses kann zwischen *Kontinuierlich* und *Jobgebunden* gewählt werden. Trotz gegenteiliger Aussage von Hewlett-Packard funktioniert nur der »Kontinuierlich«-Modus. Somit ist es leider nicht möglich, mit mehreren Rechnern gleichzeitig die Print-Server von HP unter OS/2 Warp direkt zu nutzen, da

im »Kontinuierlich«-Modus der Zugang von dem Rechner, der zuerst die Verbindung etabliert hat, blockiert wird. Ungeachtet dessen kann jeder aber den Status des Drucker mit JetAdmin abfragen. Wird bei nur einem Rechner im Netz der HP-Netzwerkanschluß-Treiber installiert, kann dieser den Drucker benutzen und auf konventionelle Weise als Netzwerkressource zur Verfügung stellen. Dies stellt aber meiner Meinung nach nur einen zweitklassigen Behelf dar. Eine Nachbesserung seitens Hewlett-Packards ist hier dringend notwendig.

In den Einstellungen *Anschlußoptionen* des Netzwerkanschlusses lassen sich weitere Optionen einstellen: so kann beispielsweise das Bedienfeld des Druckers gesperrt werden, sodaß niemand Änderungen am Menü des Druckers abspeichern kann. Weitere Informationen können Sie der Online-Hilfe entnehmen.

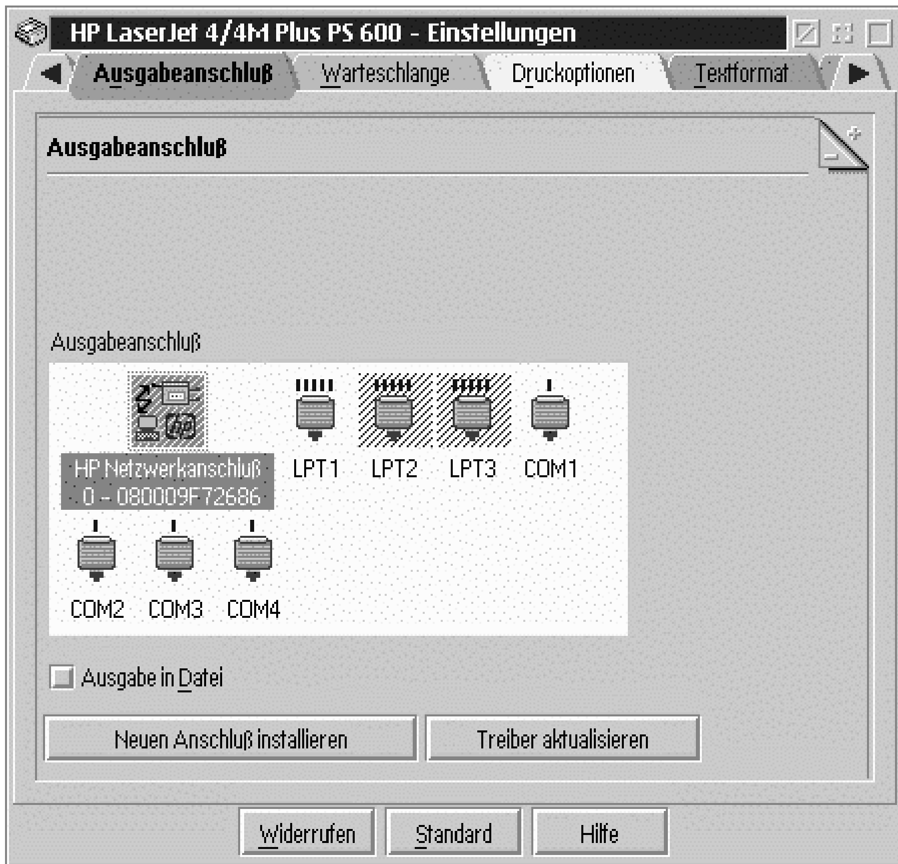


Abb. 11.36: Der HP-Netzwerkanschluß wurde als Ausgabeanschluß ausgewählt.

11.7 Remote Services Management

von Oliver Mark

Die Remote Services, oder auch Remote Support für OS/2, oder auch PolyPM/2 lite, dienen dazu, dem Supportteam der IBM kontrollierten Zugang zum einem OS/2-PC mittels Modem zu gewähren. Wie man an den verschiedenen Namen erkennen kann, steckt dahinter eine Strategie (Remote Services), ein IBM-Produkt (Remote Support für OS/2) und ein Basisprodukt, aus dem das IBM-Produkt entstanden ist (PolyPM/2 lite). Funktionell entspricht das »Remote Service Tool« in etwa dem »Remote Workstation Control« aus den »Remote Access Services«; mit der Einschränkung, daß dies hier lediglich über Modemleitung genutzt werden kann und daß der Managercode (also der Gegenpart zum ausgelieferten Client) ausschließlich dem Support der IBM zur Verfügung steht.

Zusätzlich zu dem grafischen Möglichkeiten der Überwachung existieren Hilfsmittel, um Files auszutauschen und eine Chat-funktion, um sich im Dialog per Text zu verständigen.

Um die notwendigen Sicherheitsaspekte für Benutzer zu gewährleisten, sind verschiedene Mechanismen implementiert:

- Nur das IBM-Supportteam (International Software Solutions) hat Zugang zu der Manager-Komponente des Produktes.
- Der Remote-Support-Client kann nur mittels des Remote-Support-Manager angesprochen werden. Andere Produkte (DCAF, Netfinity, PolyPM) haben keine Zugriffsmöglichkeit auf den Remote-Support-Client.
- Jeder Zugriff auf die Maschine benötigt eine zugängliche Modemverbindung und das interaktive Einverständnis des lokalen Benutzers.

Über die IBM-International-Software-Solutions-Abteilung können zusätzlich weitere Varianten des Produktes für verschiedene Verwendungszwecke erworben werden:

- R.S.M. lite: Zusätzlich zur Modemverbindung sind hier NetBIOS, TCPIP, IPX oder ISDN-Verbindungen möglich.
- R.S.M. advanced: Zusätzlich zu R.S.M. lite, Verbindungsmöglichkeiten über X.25 und APPC/APPN (SNA backbones).
- R.S.M. professional: Zusätzlich zu R.S.M. advanced ist hier eine API-Schnittstelle implementiert, die Entwicklern eigene Kommunikation über R.S.M. ermöglicht, so zum Beispiel zur Realisierung einer eigenen Softwareverteilungslösung.

11.7.1 Voraussetzungen

Um eine Verbindung zum IBM Support aufbauen zu können, brauchen Sie:

– ein Modem und ein Telefon, idealerweise an einer Leitung (NF-Dose der Telekom)

– eine korrekt installierte und konfigurierte Remote-Support-Installation.

– ein Problem, das vom Support bearbeitet werden muß.

Haben Sie die beiden ersten Bedingungen erfüllt, und die unwahrscheinliche Situation tritt ein, daß auch die dritte Bedingung zutrifft, sollten Sie den IBM-Support über eben dieses Telefon anrufen. Dann kann sich der Mitarbeiter des Supportes nach telefonischer Klärung des Problems direkt auf Ihr Modem und Ihren PC zuschalten. Dies hat den Vorteil, daß Sie hier die meist kostengünstige Verbindung über die Support-Telefonnummer weiterbenutzen können.

Haben Sie die obigen Voraussetzungen nicht, so rufen Sie den Support an, schildern das Problem und hinterlassen die Nummer, auf die das Modem geschaltet ist. Der Support-Mitarbeiter der IBM ruft dann diese Nummer an und aktiviert eine Callback-Funktion im Remote Support, und dies könnte durchaus teurer für Sie werden.

11.7.2 Installation

Der Remote Support können Sie im Rahmen der Bonuspak-Installation bei einer Komplettinstallation mit auswählen oder mittels Selektiver Installation als Bonuspak-Komponente nachinstallieren.

Die Installation meldet sich mit dem Eingangsfenster, indem Sie die Installation oder das Löschen des Produktes auswählen können. Wählen Sie hier *Install*.



Abb. 11.37: Installationsfenster

Als erstes müssen Sie sich als Benutzer eintragen und das Installationslaufwerk eingeben. Tragen Sie einen sinnvollen Benutzernamen ein, denn Sie finden Ihn später als Eintrag in Logfiles

wieder. Das Installationslaufwerk darf kein komprimiertes Laufwerk sein, kann aber sonst jedes FAT oder HPFS-Laufwerk sein. Die Installation selbst benötigt dort lediglich 1 Mbyte Platz.

Abb. 11.38: Benutzereintrag

Klicken Sie danach auf den Schalter *Install* und ein Prozeßanzeigefenster erscheint, das Ihnen anzeigt, wieweit die Installation fortgeschritten ist.

Erscheint dort der Eintrag *Installation complete*, können Sie das Fenster mit dem Schalter *Quit* beenden und kommen auf das Konfigurationsfenster zur Einstellung Ihres Modems. Wählen Sie dort die entsprechende Schnittstelle für Ihr Modem aus. Die Baudrate passen Sie entsprechend Ihrem Modem an; unterstützt Ihr Modem mehr als 14.400 bps, stellen Sie diesen Eintrag auf 38.400 bps.

In der Auswahlliste der Modems wählen Sie Ihr Modem aus. Ist es dort nicht aufgeführt, wählen Sie den Eintrag *Default* aus. Danach bestätigen Sie Ihre Einstellungen mit OK.

Sie kommen dann wieder auf das erste Installationsfenster zurück, von wo Sie die Installation mittels *Exit* beenden. Das Programm weist Sie dann auf einen durchzuführenden Systemabschluß hin.

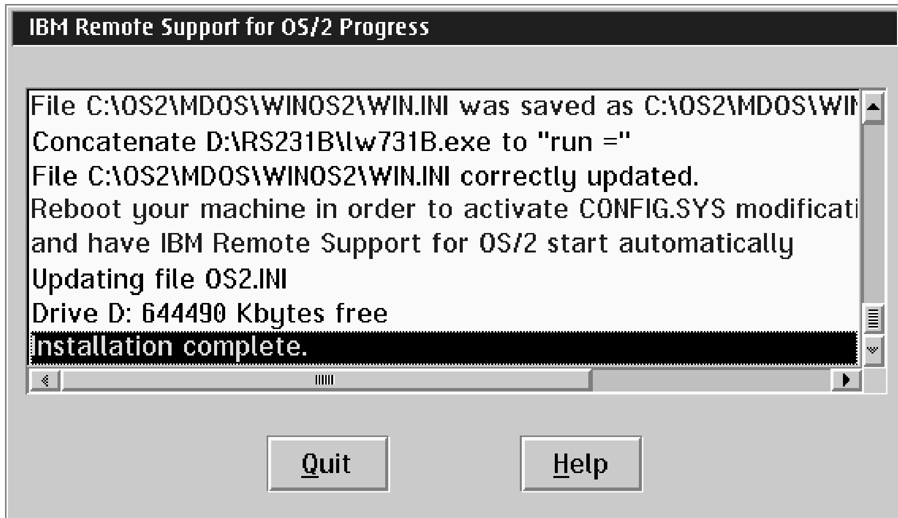


Abb. 11.39: Systemabschluß

Das Installationsprogramm hat nun entsprechend Ihrem Laufwerksangaben folgende Einträge in der »config.sys« vorgenommen:

```
LIBPATH=.............;C:\RS231B;....................
DEVICE=C:\RS231B\ECRDRV.SYS
DEVICE=C:\RS231B\RSMVDWO.SYS
CALL=C:\RS231B\PPS731B.EXE
```

Bevor Sie nun das erste Mal Remote Support benutzen können, sollten Sie unbedingt einen Systemabschluß durchführen.

Benutzung des Remote Support Client

Klicken Sie auf *Client*, wird die grafische Oberfläche gestartet.



Abb. 11.40: Remote Support Client

Dort haben Sie unter den jeweiligen Menüpunkten die Möglichkeit, den Client zu rekonfigurieren, den Client in einen Schlafmodus zu versetzen (siehe unten), ihn aus diesem wieder zu erwecken, einen eingehenden Anruf anzunehmen und alle Logfiles anzusehen.

Modem-Setup

Unter SETUP finden Sie Einstellungen für das Modem und das Verhalten des Remote-Support-Clients. Wählen Sie dort *Modem* aus, um die während der Installation gemachten Einstellungen zu ändern oder ein nicht eingetragenes Modem manuell zu konfigurieren.

Auch hier können Sie die serielle Schnittstelle Ihres Modems auswählen, und mittels Schieberegler die Übertragungsgeschwindigkeit einstellen. Für ein analoges Modem, das eine schnellere Übertragungsrate als 14.400 bps unterstützt, stellen Sie hier 38.400 bps ein. Unter *Modem* haben Sie die Auswahlliste der definierten Modems.

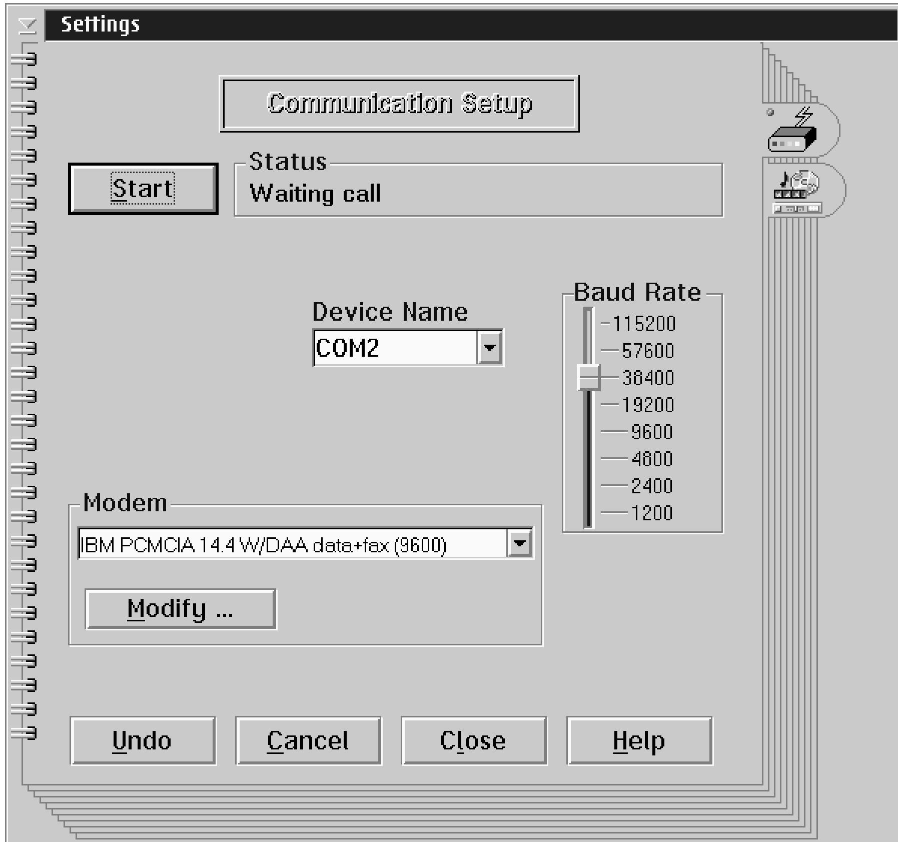


Abb. 11.41: Kommunikationseinstellungen

Mittels des Schalters *Modify* können Sie nun eine neue Modemdefinition anlegen oder eine bestehende Definition ändern und dort die entsprechenden Modem-Strings angeben. Dies kann unter Umständen notwendig sein, wenn Ihr Telefonanschluß noch kein Tonwahlverfahren unterstützt. Hierzu sollten Sie auf alle Fälle das Handbuch des Modems bereitlegen.

In den einzelnen Feldern sollten Sie nur nach Bedarf und nach Anweisung des Modemhandbuches Änderungen vornehmen:

– Im Titelfeld tragen Sie den Namen Ihres Modems ein.

– Der *Initialisierungsstring* dient der Grundinitialisierung Ihres Modems (siehe Handbuch)

– Der *Programmingstring* konfiguriert Ihr Modem für ansynchrone Kommunikation.

– Der *Waiting Call String* konfiguriert das Modem, Autoanswer-Betrieb zu aktivieren.

– Der *Dialing String* dient zur Anwahl eines Modems. Für Tonwahl belassen Sie hier AT DT, für die vielerorts noch verbreitete Pulswahl ändern Sie hier den Eintrag auf AT DP.

Die Checkboxen am unteren Rand des Fensters dienen zu Optimierung der Verbindung:

– *Use Line Speed* dient zur automatischen Erkennung einer Übertragungsgeschwindigkeit des anwählenden Modems. Schalten Sie diese Option nur ein, wenn Ihr Modem keine Übertragungsraten über 14.400 BPS uterstützt.

– *CTS/RTS Flow Control* aktiviert die Hardwareflußkontrolle, das heißt das Modem selbst kann die Übertragungskontrolle steuern. Dies unterstützen normalerweise alle Modems. Deaktivieren Sie diesen Punkt nur, wenn Sie einen eindeutigen Hinweis im Handbuch des Modems finden.

– *Fast Modem Setup* ermöglicht es dem Modem, die oben definierten Kommando-Strings als komplettes Kommando zu übertragen und nicht als einzelne Charakter. Sollten Sie trotz korrekter Strings Fehlermeldungen bei der Anwahl erhalten, sehen Sie in Ihrem Modemhandbuch nach, ob Ihr Modem diesen Betrieb nicht unterstützt.

– *Continous Watching Modem* veranlaßt das Remote-Support-Tool, in regelmäßigem Abstand einen eingehenden Ruf zu erkennen. Auch diesen Modus unterstützen nicht alle Modems. Außerdem sollte der Supportfall eine Ausnahmesituation darstellen, so daß Sie den eingehenden Ruf manuell annehmen können.

Sollten Sie weiterhin Probleme mit dem Modem haben, entspricht es wahrscheinlich nicht dem Hayes-Standard für Modem-Kommunikation, der mindestens folgende Kommandos voraussetzt (vergleichen Sie gegebenenfalls mit dem Handbuch des Modems):

```
AT &F, AT Z, E1, X1, Q0, &C1, &D2, &Kx, <CR>
```

Mittels des Schalters *Terminal* können Sie Remote Support auch als einfaches Terminalprogramm benutzen, um die Strings zu überpüfen.

Speichern Sie die gemachten Änderungen mit *Save*. Remote Support legt dann eine Datei »mdm.usr« an, in der Ihre persönlichen Angaben eingetragen werden. Die Standardangaben in der Datei Mdm.Lst sollten Sie keinesfalls (auch nicht manuell) ändern.

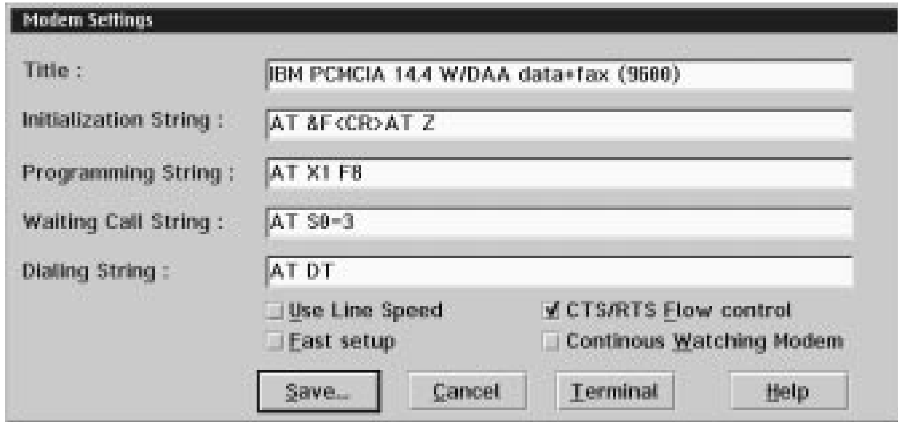


Abb. 11.42: Modem Settings

Modem-Miscellaneous

Hier haben Sie die Möglichkeit, Soundunterstützung für Remote Support zu aktivieren. Hierzu können Sie im oberen Teil den Sound der Hilfe-Bubbles aktivieren. Wie Sie den einzelnen Bubbles Sounds zuweisen, sehen Sie im folgenden. Mit der Auswahl *Automatic* erzeugen Sie automatisch bei die Anzeige der Hilfe-Bubble, wenn Sie mit dem Cursor im entsprechenden Feld sind. Mit der Auswahl *Right Button* müssen Sie mit der rechten Maustaste in das Feld klicken, für das Sie die Hilfe angezeigt bekommen wollen.

Im unteren Teil können Sie den einzelnen Aktionen von Remote Support Sounds zuweisen. Hierzu wählen Sie die entsprechende Aktion und geben Sie im Feld *Wave File* den Sound an. Mit *Play* können Sie sich die Sounds vorab anhören.

Wenn Sie einem Hilfe-Bubble einen Sound zuordnen wollen, so rufen Sie die entsprechende Hilfe auf; entweder mittels automatischer Aktivierung oder mittels rechter Maustaste. Wenn Sie die entsprechende Hilfe-Bubble aktiv haben, drücken Sie die Taste [F2]. Links oben in der Hilfe-Bubble bekommen Sie nun den Namen dieser Bubble angezeigt. Notieren Sie sich diesen Namen ohne die erste Stelle. Erzeugen oder kopieren Sie sich den gewünschten Sound in das RS231B-Unterverzeichnis Ihrer Festplatte. Eine Hilfe-Bubble mit dem Namen B004A0050 würde dann einen Sound mit dem Namen 004A0050.Wav bedeuten. Von nun an wird dieser Sound beim Aufruf dieses Bubbles abgespielt.

Rufannahme

Haben Sie Verbindung zum Supportteam telefonisch aufgenommen, können Sie, wie eingangs beschrieben, entweder über eine gemeinsame Leitung die Kontrolle übergeben oder über die Rückrufoption vom Supportmitarbeiter aktivieren lassen.

Wenn Sie der Supportmitarbeiter auffordert, die Kontrolle an ihn zu übergeben, klicken Sie auf *Answer* in der Menüleiste. Die Kontrolle, inklusive Maus und Tastatur, geht an den Supportmitarbeiter über, Sie haben nun keine Eingriffsmöglichkeit mehr (außer Sie schalten das Modem aus). Bei geteilter Leitung ruft Sie der Supportmitarbeiter an und aktiviert den Rückruf.

Suspend/Restart

Haben Sie Automatische Rufkontrolle (»Continuous watching modem«) aktiviert, haben Sie mittels *Suspend* die Möglichkeit dies dynamisch zu unterbrechen und mittels *Restart* wieder aufzunehmen. Dies ist dann notwendig, wenn Sie *continuous watching* aktiv haben und kurzfristig ein anderes Kommunikationsprogramm auf dem Modem verwenden wollen.

12 ISDN

ISDN steht nicht für »Is Sowas Denn Nötig«, sondern für »Integrated Services Digital Network«.

Das Diensteintegrierende Digitalnetz (so sein Name im »Bundespost-Deutsch«) war bei seiner Einführung die Lösung jeglicher Kommunikationsprobleme: Eine bis dahin (2.400 bit/s) nicht gekannte Geschwindigkeit (64.000 bit/s), mit Komfortmerkmalen (Makeln, Konferenz, Rufweiterschaltung, Diensterkennung, Rufnummerübermittlung...) und vielen weiteren Vorteilen, von besserer Sprachqualität und schnellerem Verbindungsaufbau bis hin zu mehreren Rufnummern für verschiedene Geräte.

12.1 Die Technik von ISDN

1-2-3 = 1 Anschluß – 2 Leitungen – 3 Rufnummern

Ein ISDN-Standardanschluß bietet »2 Amtsleitungen« oder besser Nutzkanäle, die B-Kanäle und einen D-Kanal, der hauptsächlich dazu genutzt wird, Protokollaufgaben zu übernehmen. So geht über den D-Kanal beispielsweise die Information ein, daß ein Teilnehmer mit der Nummer 0234/5678 mit einem ISDN-Telefon den Teilnehmer 0345/6789 sprechen möchte. Ein entsprechendes Endgerät, das einen Telefonanruf entgegennehmen kann und auf die entsprechende Nummer (sog. MSN) konfiguriert ist, klingelt nun. Sobald der Hörer abgenommen wird, wird über den B-Kanal eine Verbindung aufgebaut.

ISDN ist vollkommen digital. So wird die Sprache direkt vom Telefon oder einem entsprechendem Gerät (a/b Wandler, ISDN Telefonanlage) in digitale Informationen übersetzt. Rufen Sie einen Nicht-ISDN-Teilnehmer an, setzt die Vermittlungstelle des Angerufenen die digitalen Informationen wieder in analoge um, andernfalls geht die digitale Information bis zum Empfänger. Dies verbessert die Qualität von Sprache und Datenübertragung.

Ruft Sie umgekehrt jemand von einem analogen Anschluß an, wird in der Vermittlungsstelle eine Digitalisierung vorgenommen. Anrufe von analogen Anschlüssen unterstützen die ISDN-Dienstmerkmale Rufnummernübermittlung und Diensterkennung nicht. Die Rufnummernübermittlung wäre technisch machbar, die Telekom scheut dies jedoch wegen der rechtlichen Unsicherheit.

Die Telekom nutzt für ISDN im allgemeinen das normale Leitungsnetz, d.h., Kupferkabel, daher müssen entsprechende Vorkehrungen getroffen werden, um eine verlustfreie Übertragung zu ermöglichen. Unter anderem aus diesem Grund wird an ihrem ISDN-Hauptanschluß ein NTBA (zu deutsch etwa Netzabschluß) installiert.

Die normalen Kupferkabel haben den Nachteil, das sie sich gegenseitig stören, um sie herum baut sich, sobald eine Spannung anliegt, ein Magnetfeld auf, und dieses induziert sich auf die anderen Leitungen. Grundsätzlich hat man dieses recht gut durch Abschirmungen und entsprechende Abstände im Griff. Bei höheren Frequenzen, im Falle von ISDN 8 kHz, treten jedoch Echo-Effekte auf, sofern keine technischen Vorkehrungen getroffen werden. Diese Aufgabe übernimmt auf Anschlußseite der NTBA.

Am NTBA kommen 4 Adern an (beim analogen Anschluß in der Regel nur 2, genannt »a« und »b« – daher auch der Begriff a/b-Wandler), in der Regel werden diese Adern verschraubt, immer öfter findet man jedoch auch Steckverbindungen in Form eines sog. Westernsteckers, der einfach in den NTBA gesteckt wird. Ein falscher Anschluß dieser Adern hat bei unserer heimischen Anlage schon dazu geführt, daß ca. 10.000 ISDN-Anschlüsse tot lagen – der arme Telekomtechniker kam schließlich mit Verstärkung und hinterließ einen dann funktionierenden Anschluß.

Nicht nur aus diesem Grunde sollten Sie das Anschließen des NTBA lieber einem Fachmann überlassen.

Nach dem NTBA können Sie nur noch wenig falsch machen, er stellt zwei Westernstecker zur Verfügung, an die sie entweder direkt ihre ISDN-Karte anschließen können oder aber eine Telefonanlage oder eine Verteilung. An einem normalen ISDN-Anschluß sollten nicht mehr wie 12 Anschlüsse vorhanden sein, auch sollten Sie nicht mehr wie 8 Geräte anschließen, dabei sind jedoch nur vier ISDN-Telefone möglich. Diese Begrenzung ergibt sich durch die begrenzte Stromversorgung des NTBA. Jede Anschlußdose muß – ähnlich wie beim SCSI-Bus – terminiert werden, d.h., ein sog. Abschlußwiderstand wird angebracht.

Hinweis: Im Gegensatz zum analogen Telefon (dort stellt die Vermittlungsstelle die Stromversrgung sicher) liefert bei ISDN der NTBA die Stromversorgung. Bei Stromausfall steht ihnen der ISDN-Anschluß in der Regel daher nicht zur Verfügung! Beachten Sie dies in Notfällen!

Die Vermittlungsstelle stellt dann die Stromversorgung für ein ISDN-Telefon sicher.

12.1.1 Die ISDN-Protokolle

ISDN übermittelt über den D-Kanal verschiedene Daten, die die jeweiligen Endgeräte nutzen können, angefangen von der aktuellen Zeit, über die Telefonnummer bis hin zum Dienstmerkmal, daher können Sie z.B. ISDN-Karte und Telefon dieselbe Rufnummer geben ohne befürchten zu müssen, daß Sie oder ein Anrufer »falsch verbunden werden«.

12.1 2 Wichtige Dienstkennungen des ISDN:

– 64 kBit/s-Übermittlungsdienst – ISDN Datenübertragung

Es handelt sich dabei um nicht anderes als Datenübertragung über ISDN. Beide Seiten müssen untereinander kompatibel sein – dies ist vor allem von Bedeutung, wenn Sie Standards nutzen, die nicht definiert sind, wie zum Beispiel Datenkompression oder Kanalbündelung.

– 3,1 kHz-a/b-Übermittlungsdienst – analoge Datenübertragung

Wenn Sie einen a/b-Wandler verwenden, wird dieses Dienstmerkmal weitergegeben. An diesem a/b-Wandler, der auch in Ihrer Telefonanlage intgriert sein kann, können sowohl »alte« Telefone, wie auch Modem oder ein Fax der Gruppe 2/3 hängen.

– Telefonie 3,1 und 7 kHz

Sobald einer der Gesprächsteilnehmer ein Telefon hat, das die 7-kHz-Übertragung nicht unterstützt (das sind alle analogen und viele ISDN-Anschlüsse, an denen keine entsprechenden ISDN-Telefone angeschlossen sind), wird die Sprache mit einer Auflösung von 3,1 kHz übertragen. Zum Vergleich: eine CD hat eine Auflösung von bis zu 48 kHz, selbst ein ISDN-Telefon klingt also nie so gut wie eine CD, die Sprache ist nur weniger »dumpf«.

– Telefax Gruppe 2/3

Ein Fax von einem »normalen« Fax sollte dieses Dienstmerkmal haben.

BTX (alt): T-Online über die analogen Zugänge.

BTX (neu mit 64.000 bit/s): T-Online mit ISDN-Zugang.

Bildetelefonie

Hier müssen Sie beide Nutzkanäle (B-Kanäle) opfern. Entsprechende Hardware vorausgesetzt, hören Sie jetzt nicht nur Ihren Partner, sondern Sie sehen ihn auch.

Das dafür zuständige Protokoll ist DSS1 (Digital Signaling Standard). DSS1 ist europäisch einheitlich, man spricht daher auch von einem Euro-ISDN-Anschluß oder EDSS1. Vor DSS1 gab es verschiedene nationale Protokolle, in Deutschland war dies 1TR6 (TR=Technische Richtlinie), man spricht dann vom deutschen oder nationalen ISDN. Im Umfeld großer Telefonanlagen gibt es außerdem 12TR7, es setzt sich aus DSS1 und 1TR6 zusammen und kann um herstellerspezifische Merkmale erweitert werden. 1TR6, das nationale ISDN, hat keine Zukunft, es wird im Jahr 2000 von der Telekom abgeschaltet. Alle neuen Anschlüsse verwenden daher DSS1.

Im nationalen ISDN kennt man Endgeräteauswahlziffern (EAZ), d.h., jedem ISDN-Anschluß wurde ein Nummernblock zugeteilt, den man wiederum entsprechenden Geräten zuweisen konnte.

Beispiel: 02 34 / 12 34 5 – 0 ISDN-Telefon

– 1 Analoges Fax

– 2 ISDN Datenübertragung

...

– 9 ...

Das europäische ISDN kennt dieses System nicht, es benutzt »Multiple Subscriber Numbers« (MSN), diese Nummern (in Deutschland normalerweise drei) können jedem beliebigen Gerät zugeordnet werden. Wird z.B. eine Nummer mehreren verschiedenen Telefonen zugeordnet (Hotline), bekommt jeweils derjenige das Gespräch, der abnimmt. Auch dieselbe Nummer für ISDN-Mailbox und normales Telefon ist denkbar, der ankommende Ruf beinhaltet neben der Rufnummer, für die er bestimmt, ist auch das Dienstmerkmal, also digitale Datenübertragung oder Telefonie. Es ist sogar denkbar, daß gleichzeitig ein Daten- und ein Telefonanruf auf derselben Nummer eingehen und entgegengenommen werden können...

12.1.3 The need for Speed – ISDN ist schnell

Muß es schnell gehen, unterstützen viele Karten auch die Kanalbündelung, d.h., beide B-Kanäle werden zur Datenübertragung genutzt. Das bedeutet einen Datendurchsatz von bis zu 128 Kbit/s oder ca. 16.000 Zeichen/s. Derzeit ist für ISDN noch keine Datenkompression definiert, oftmals werden in der Werbung traumhafte 300 Kbit/s beworben. Dies ist jedoch ein theoretischer Wert, der höchstens bei typgleichen Karten mit gleicher Software und optimalen unkomprimierten Daten (z.B. Text) erreicht wird.

Sie sollten umgekehrt daher auch nicht auf die Werbung der Modemhersteller hereinfallen, die Ihnen ein Modem, das 33,1 kBit/s unkomprimiert überträgt, als schneller als ISDN anbieten. ISDN wird bei den hauptsächlich bereits komprimierten Daten immer schneller sein, selbst mit nur einem Nutzkanal.

12.1.4 Welcher Anschluß?

ISDN-Mehrgeräteanschlüsse gibt's in drei Variationen: Einfach-, Standard- und Komfortanschluß.

– Der Einfachanschluß

Ein ISDN-Anschluß light, eine Nummer, einen Nutzkanal, keine Komfortmerkmale. Kostet derzeit DM 41,-, eine uninteressante Variante.

– Der Standardanschluß

Dies ist wie der Name sagt Standard, ein Anschluß, zwei Nutzkanäle, drei Rufnummern. Kostet derzeit DM 46,-, wenn Sie die Komfortmerkmale wie Konferenz, Makeln, umstecken am Bus, Übertragung der Gesprächskosten am Verbindungsende nicht benötigen, reicht diese Anschlußvariante.

– Der Komfortanschluß

Hier bekommen Sie für DM 51,- wie beim Standardanschluß, einen Anschluß, zwei Nutzkanäle und drei Rufnummern, außerdem die Komfortmerkmmale. Bei kleinen Telefonanlagen ist dies schon fast Pflicht, denn nur so kann man korrekt abrechnen (Übertragung der Gesprächskosten am Verbindungsende). Eine Überlegung wert ist außerdem die detaillierte Abrechnung der Telekom, die kostet pro Monat DM 5,-. Sogar der Komfortanschluß ist günstiger wie ein gleichwertiger analoger Doppelanschluß.

12.2 ISDN mit dem PC

ISDN mit dem PC – viele verstehen darunter Datenübertragung, doch eine ISDN-Karte im PC kann mehr. Ihre ISDN-Karte kann beispielsweise mit entsprechender Hard-/Software Anrufbeantworter, Echtzeituhr, Fax, Modem, Netzwerkkarte, Komforttelefon, Bildtelefon und vieles andere mehr werden. Im folgenden möchte ich Ihnen nun die einzelnen Möglichkeiten aufzeigen.

12.2.1 ISDN und OS/2

OS/2 ist bei Insidern als das DFÜ-Betriebssystem schlechthin bekannt.

OS/2 bietet mit seinem stabilen Multitasking ideale Voraussetzungen, um anspruchsvolle DFÜ-Anwendungen von Mailboxen mit vielen Lines über Faxabruf bis hin zum integrierten Telefonmanagementsystem.

Viele Mailboxen von kleinen Hobbymailboxen bis hin zur Firmensupportmailbox laufen unter OS/2.

Voraussetzungen

Sie benötigen vom Hersteller Ihrer ISDN-Karte eine aktuelle CAPI für OS/2, eine DOS- oder Windows-CAPI hilft ihnen nicht weiter. Alle namhaften Hersteller von ISDN-Hardware bieten OS/2-Support in Form aktueller OS/2-CAPI-Versionen.

Sofern Sie über speziell für ISDN vorbereitete Software verfügen, war's das bereits. Die Software erkennt automatisch die vorhandene CAPI und nutzt diese.

Viele Programme werden jedoch im Ausland geschrieben, dort ist ISDN oftmals ein Fremdwort, aber auch hier kann geholfen werden. Es gibt ISDN-Software, die sich auf die CAPI aussetzt und der Software, die kein ISDN »versteht«, etwas »verständliches« emuliert – einen COM-Port. So können selbst älteste DOS-Anwendungen auf 64 Kbit/s beschleunigt werden.

Wie bereite ich nun meinen Rechner für den ISDN-Einsatz vor?

Viele CAPIs werden über normale OS/2-Treiber realisiert, dabei sollte beachtet werden, daß OS/2 etwas restriktiver ist, wenn z.B. Interrupts doppelt belegt werden, unter reinem DOS kann dies funktionieren (muß nicht), OS/2 wird dies mit einem Absturz quittieren. In diesem

Fall konfigurieren Sie ihren Rechner so, daß keine Interrupts doppelt belegt werden. Hilfreich kann hier die [ALT]+[F2] Funktion sein, die ihnen beim Booten die geladenen Treiber anzeigt.

Sobald sie den Rechner umkonfiguriert haben, sollten Sie mit [ALT]+[F1] den Rechner mit voller Hardwareerkennung booten lassen. Hängt er sich wieder auf, rufen Sie mit [ALT]+[F1] die Kommandozeile auf und entfernen Sie zunächst die ISDN-Treiber und Einträge aus der Congig.Sys. Anschließend sollte das System booten, installieren Sie dann die CAPI erneut mit den korrekten Parametern (Interrupts und Speicheradressen).

Hier ein paar Tips:

Der zweite IDE-Port belegt gerne den Interrupt 15, viele ISDN-Karten nutzen diesen standardmäßig.

Deaktivieren Sie diesen zweiten IDE-Port im BIOS.

IRQ 7 kann unter OS/2 für ISDN genutzt werden, OS/2 braucht (bei Standardeinstellung) keinen Interrupt für die parallele Schnittstelle.

Was ist eine CAPI?

CAPI bedeutet »Common Application Programming Interface«. Die CAPI wurde in Deutschland erfunden, die Hersteller von ISDN-Hardware erkannten, das sie eine gemeinsame Basis schaffen mußten, auf der jede Software laufen mußte. Die Hardware ist sehr unterschiedlich, so stellte man einheitliche Softwarefunktionen bereit, die es ermöglichen, unabhängig von der jeweiligen Karte für CAPI geschrieben Software einzusetzen.

CAPI 1.0 wurde für das deutsche ISDN (1TR6) entwickelt und ist heute nahezu bedeutungslos. Mit Einführung von Euro-ISDN (DSS1) wurde eine Anpassung der CAPI notwendig, gleichzeitig sollte die CAPI kompatibel bleiben. CAPI 1.1 unterstützt nunmehr z.B. MSN und andere »Neuheiten« von DSS1. CAPI 2.0 wurde voll und ganz mit Blick auf das Protokoll DSS1 entwickelt, sie bietet eine einfachere Installation und mehr Funktionen. Mit CAPI »MS« (wenn man sie so nennen kann), versucht Microsoft hier einen eigenen Standard.

Wie konfiguriere ich die CAPI?

Für OS/2 sind derzeit nur CAPIs nach der Spezifikation 1.1 erhältlich. In Deutschland gibt es aus historischer Sicht zwei ISDN-Protokolle, das nationale (1TR6) und das europäische (DSS1), im Normalfall stellen Sie hier DSS1 als Protokoll ein.

Weiterhin müssen Sie einstellen, welche EAZ welcher MSN entspricht, 0 = 123450 1=123451 2=123452. Dies ist wichtig, um die CAPI kompatibel zum neuen Standard zu machen.

Ob die CAPI korrekt funktioniert, können Sie mit Z-O-C testen, ZOC ist ein Terminalprogramm für OS/2, das von Haus aus direkt auf die CAPI zugreifen kann, also keine COM-Port-Emulatoren braucht. Gelingt ein korrekter Verbindungsaufbau, haben Sie korrekt konfiguriert.

Unter OS/2-DOS und Win-OS/2 sollten Sie die Speicheradressen vorsichtshalber ausschließen.

Öffnen Sie das Kontextmenü, wählen Sie *Einstellungen|Sitzung|DOS-Einstellungen|MEM_EXCLUDE_REGIONS* und tragen Sie dort die entsprechende Speicheradresse ein.

Empfehlenswerte Programme für ISDN-DFÜ mit OS/2:

Textmodus-Terminalprogramme: TE/2

PM-Terminalprogramme: Z-O-C oder LiveWire

Mailboxsoftware: Blinkley/2, LORA/2, Maximus/2, CL

BTX-Software: Opalis BTX bzw. IBM Opalis (wird mitgeliefert)

Faxsoftware: Faxworks, PM Fax

Grundsätzlich sind dies nur Vorschläge, diese müssen nicht Ihre Bedürfnisse oder Ihren persönlichen Geschmack treffen, sehen Sie sich ruhig um, vollständig ist die obige Liste nicht.

Viele Programme laufen mit unveränderten Parametern bereits sehr gut auf einer richtig konfigurierten CAPI. Die ISDN-Fähigkeiten können Sie eigentlich erst mit einem Betriebssystem wie OS/2 richtig nutzen. Denn dort, wo DOS nur eine Anwendung ausführt, Windows 3.11 schon bei einer DFÜ-Anwendung Probleme bekommt und nur mit viel Aufwand mehr wie eine DFÜ-Anwendung passabel ausführt, fangen die Vorteile von OS/2 erst an.

Der COM-Port-Emulator

CFOS/2, Micado und andere alle erfüllen einen Zweck, sie helfen »alter« Software auf die neuen schnellen ISDN-Beine. Die beiden ISDN-Kanäle werden zu COM 3 und 4, das versteht jede Software. Viele DFÜ-Programme lassen sich einfach auf COM 3 oder 4 umstellen, setzen Sie die Schnittstellengeschwindigkeit von z.B. 38.400 auf 64.000 oder 115.000, auf jeden Fall sollte der Wert >=64.000 sein, sonst kann es sein, daß Sie nicht die volle ISDN-Bandbreite nutzen.

Sollten Sie noch auf DOS-DFÜ-Software setzen, haben Sie jetzt einen guten Grund, auf OS/2-Software umzusteigen...

12.3 Die verschiedenen Datenübertragungsprotokolle in ISDN

Das bekannteste ist sicherlich X.75, es bietet volle ISDN-Geschwindigkeit.

V.110 gibt es in verschiedene Abwandlungen: 9.600, 19.200, 38.400 Baud. Diese Protokolle wurden entwickelt, um die Belastung der Hardware gering zu halten. Dieses Protokoll gehört aber zum alten Eisen, es sei denn, Sie nutzen CompuServe, dort wird es noch benutzt. Daneben gibt es noch weitere sogenannte B2-Protokolle, z.B. für BTX, diese werden Ihnen jedoch nur sehr selten begegnen.

12.4 Wie verwende ich meine alte Software mit ISDN?

Grundsätzlich laufen alle »alten« Programme mit dem COM-Port-Emulator. Fax-Software, bestimmte Anrufbeantworter und spezielle Zugangssoftware zu Onlinediensten wird in der Regel nicht einfach einzusetzen sein.

12.5 Fax via ISDN

Grundsätzlich gibt es zwei Arten von Faxen: die große Gruppe der analogen Faxgeräte (G2- oder G3-Fax) und die sehr kleine Gruppe der digitalen Faxgeräte G4.

Analoges Fax (Gruppe 2/3)	**Digitales ISDN-Fax (Gruppe 4)**
Datenübertragung mit bis zu 14.400 Bit/s	Datenübertragung mit 64.000 bit/s
Auflösung bis 196 dpi	Auflösung 400 dpi
Sehr weit verbreitet	Kaum verbreitet
kostengünstige Endgeräte	Sehr teure Endgeräte
Heim- und Geschäftsbereich	Highend-Bereich
Emulation durch Modem oder ISDN-Karte möglich	Emulation bei sehr teuren ISDN-Karten möglich

Die beiden Gruppen sind nicht miteinander kompatibel.

Ein von Ihrer ISDN-Karte verschicktes Fax ist kein G4-Fax, sollten Sie tatsächlich ein G4-Fax als Gegenstelle haben, wird dieses nicht abnehmen und eine Faxübertragung nicht zustandekommen.

Faxe sollten Sie grundsätzlich mit einem Modem oder einem Fax-Gerät verschicken oder empfangen. Dies ist schneller als die Emulationen von ISDN-Karten, die meistens nur 9.600 Bit/s übertragen, während ein modernes Modem oder Fax 14.400 Bit/s schafft.

Das oft propagierte Gruppe-4-Fax (G4) (oder auch ISDN-Fax) ist derzeit kaum verbreitet, die entsprechende Hardware ist einfach zu teuer. Sie werden daher kein G4-Faxprogramme finden. Faxworks auf einem COM-Port-Emulator einzusetzen, hilft wenig, denn die Gegenseite ist meistens ein Fax-Gerät. Dort wird der Anruf dank Diensterkennung nicht ankommen, auch nicht auf einem Modem. Verwenden Sie daher zum Faxen immer ein Modem mit entsprechender Fax-Software oder eine ISDN-Karte mit angepaßter Software. Die Gegenstelle sollte immer ein entsprechendes analoges Fax/Modem sein.

Tip: Von ISDN-Karte zu ISDN-Karte übertragen Sie die Daten am besten mit einem entsprechend konfigurierten Terminalprogramm. ZOC beispielsweise kann in einen Empfangsmodus umgeschaltet werden. Statt ein Fax zu schicken, übertragen Sie einfach die Textdatei oder eine Grafikdatei. Das von Faxworks erzeugte Dateiformat ist eine Tiff-Abwandlung, mehrseitige Faxe werden in einem speziellen TIF-Format gespeichert, das z.B. IMPOS/2 lesen kann. Soll-

ten Sie nicht wissen, ob der Empfänger dieses Format lesen kann, drucken Sie einfach jede Seite einzeln auf Faxworks und schicken die Dateien einzeln oder als Archiv-Datei gepackt.

Wenn's nicht funktioniert...

1. Überprüfen Sie, ob Sie mit Ihrer ISDN-Karte ein entsprechendes analoges G2/G3-Fax emulieren und die Gegenstelle ebenfalls ein G2/G3-Fax ist.

2. Benutzen Sie ein Modem zum Faxversand. Ein analoges Modem mit 14.400 Bit/s bekommen Sie heute zum Teil schon geschenkt, wenn Sie z.B. bei Ihrer Bank einen T-Online-Zugang (BTX) in Auftrag geben, den Sie dann auch mit ISDN hervoragend nutzen können.

Anrufbeantworter für OS/2

Es gibt inzwischen eine reichhaltige Auswahl an OS/2-CAPI-Anrufbeantwortern. Voraussetzung ist meistens auch eine Soundkarte, um die aufgezeichneten Gespräche wiederzugeben bzw. Ansagen aufzuzeichnen. »ISDN-Wächter« ist ein gutes Beispiel für einen reinen OS/2-ISDN-Anrufbeantworter. Nicht nur, daß er Anrufe entgegennimmt und je nach eingehender Rufnummer eine entsprechende Ansage abspielen kann, vielmehr leitet er eingegangen Anrufe auch weiter, d.h., Sie können nach bestimmten Kriterien die eingegangen Anrufe an Ihren derzeitigen Anschluß weiterleiten. In Zukunft soll ISDN-Wächter auch eine direkte Anrufweiterschaltung beherrschen.

»Capi-Tel« ist eine Anwendung, die im Textmodus läuft, eine ebenfalls empfehlenswerte Anrufbeantworter-Software. Der Autor hat eine grafische Version angekündigt.

12.6 Online mit ISDN

Nicht jeder Onlinedienst bietet schon einen ISDN-Zugang. Falls doch, stellt er Ihnen auch entsprechende Software zur Verfügung. Beim Übergang ins Internet ist jedoch auch ISDN nicht das Allheilmittel, dort herrscht vielfach Stau, auch ISDN ändert daran nichts.

Stichwort Internet

Viele Internetprovider benutzen ebenfalls eine ISDN-Leitung zu ihrem Netzzugang, ein Flaschenhals, der bei mehreren Modems mit 14.400 Bit/s nicht unbedingt auffällt. Steigen Sie allerdings mit ISDN ein und der Provider »verliert« eine seiner beiden ISDN-Leitungen, kann dies unter Umständen langsamer sein wie zuvor. Große Provider wie EuNet, MAZ, Germany.NET (Netsurf.de) oder auch Onlinedienste wie AOL, vernetzen Ihre Netzknoten jedoch entsprechend, so daß Sie mit ISDN auch die volle Bandbreite zumindest bis zum Proxy (großer »Internet-Cache-Rechner«) oder z.T. auch darüber hinaus nutzen können. Stark belastete Server wie der von IBM oder Netscape sind meist über sehr schnelle Leitungen an die sogenannten Backbones (die Hauptleitungen des Internets) angeschlossen. Gerade die Firma IBM, die auch Internetzugänge anbietet, verfügt über ein hervoragendes Netz. In den vorgenannten Fällen können Sie mit ISDN zum Teil traumhafte Datenübertragungsraten erzielen.

Bitte bedenken Sie, daß das Internet hauptsächlich von den USA aus genutzt wird und dort die entscheidende Infrastruktur vorliegt. Sie sollten also ins Internet gehen, wenn die USA schlafen gehen. Gute Zeiten sind z.B. der frühe Morgen. In den Abendstunden müssen Sie damit rechnen, daß sich Angestellte von großen Firmen (Apple, IBM, Intel...) ins Internet begeben. Wenn man die Hit-Statistiken liest, wird man feststellen, daß unter den Adressen IBM.COM, Apple.COM oder MicroSoft.COM die meisten Leser der Internet-Playboy-Ausgabe zu finden sind :-)). Diese Auswirkungen sind bis ca. 21.00 h zu spüren, probieren geht auch hier über studieren.

12.7 Sonstige ISDN-Nettigkeiten

Die Rufnummernübermittlung läßt sich vielfältig nutzen, ob z.B. ihr Chef, der vom Firmeneigenen ISDN-Anschluß oder mit seinem D/E-Netz Handy anruft eine andere Ansage bekommen soll, wie der Freund(in), die von seinem/ihrem heimischen ISDN-Anschluß anruft. Für OS/2 gibt es eine reichhaltige Auswahl an Programmen, manche sogar mit kompletten Datenbanken, die Ihnen sofort den/die Anrufer/in bzw. das Ortsnetz liefern, aus dem der Anruf kommt. Empfehlenswerte kleine Programme sind »CAPI-Snoop« oder »CallerID«.

DOS/Win-OS/2

Sobald ich auf die DOS-Ebene wechsle, steht mir ein virtueller COM3/4 zur Verfügung, so daß auch Win-OS/2 von ISDN profitiert.

»CFOS/2« z.B. bietet dafür einen VCFOS-Treiber, der diese virtuelle Schnittstelle bereitstellt. Binden Sie ihn einfach als Treiber in die Config.Sys ein, wenn er allen Sitzungen zur Verfügung stehen soll oder, empfehlenswert, gehen Sie in das Kontextmenü des DOS- oder Windows-Programms, das ISDN-Zugriff benötigt, setzen Sie unter *Sitzung|DOS-Einstellungen|DOS-Device* den Treiber ohne *Device=* ein.

Bei Win-OS/2 ändern Sie die Einstellungen folgendermaßen: *Systemeinstellungen|Win-OS/2-Konfiguration|Win-OS/2-Einstellungen|DOS_Device...*

Win-OS/2

Unter Windows ist generell der Einsatz von DFÜ-Programmen nicht unbedingt empfehlenswert, das sehr unzureichende Multitasking, der kaum vorhandene Schutz der Programme untereinander, kann, wenn ein Programm abstürzt, das gerade auf den COM-Port zugreift, auch OS/2 Probleme machen. Warten Sie in diesem Fall, trennen Sie die Telefonverbindung (schalten Sie ihr Modem einige Male aus und ein oder stöpseln Sie den ISDN-Anschluß an Ihrer Karte aus und wieder ein) und benutzen Sie währenddessen [STRG]+[ESC], um zur Fensterliste zu gelangen. Dort können Sie das Windowsprogramm selektieren und durch drücken von [Del] beenden. Wenn nichts hilft, sollten Sie einen Warmstart durchführen. Sollten Sie ein Programm wie »Watchcat« einsetzen, das selbst unter schwierigsten Umständen das reibunslose Beenden von Anwendungen oder den Neustart des Systems ermöglicht, sollten Sie diese Software zuvor einsetzen.

Die Seiten 681 bis 699 (Anhänge A und B sowie Index)
des gedruckten Buchs entfallen in der Online-Version